Knowledge-Based Systems 194 (2020) 105596

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A review of deep learning with special emphasis on architectures, applications and recent trends☆

Saptarshi Sengupta [a], Sanchita Basak [a], Pallabi Saikia [b], Sayak Paul [c], Vasilios Tsalavoutis [d], Frederick Atiah [e], Vadlamani Ravi [f,*], Alan Peters [a]

[a] Vanderbilt University, Department of EECS, Nashville, TN 37235, USA
[b] IIT Guwahati, Department of Computer Science and Engineering, Guwahati 781039, India
[c] PyImageSearch, USA
[d] The National Technical University of Athens, School of Mechanical Engineering, Athens 15780, Greece
[e] University of Pretoria, Department of Computer Science, Pretoria 0002, South Africa
[f] Institute for Development and Research in Banking Technology, Center of Excellence in Analytics, Hyderabad 500057, India

ARTICLE INFO

Article history:
Received 4 June 2019
Received in revised form 28 January 2020
Accepted 30 January 2020
Available online 6 February 2020

Keywords:
Deep neural network architectures
Supervised learning
Unsupervised learning
Testing neural networks
Applications of deep learning
Evolutionary computation

ABSTRACT

Deep learning (DL) has solved a problem that a few years ago was thought to be intractable — the automatic recognition of patterns in spatial and temporal data with an accuracy superior to that of humans. It has solved problems beyond the realm of traditional, hand-crafted machine learning algorithms and captured the imagination of practitioners who are inundated with all types of data. As public awareness of the efficacy of DL increases so does the desire to make use of it. But even for highly trained professionals it can be daunting to approach the rapidly increasing body of knowledge in the field. Where does one start? How does one determine if a particular DL model is applicable to their problem? How does one train and deploy them? With these questions in mind, we present an overview of some of the key DL architectures. We also discuss some new automatic architecture optimization protocols that use multi-agent approaches. Further, since guaranteeing system uptime is critical to many applications, a section dwells on using DL for fault detection and mitigation. This is followed by an exploratory survey of several areas where DL emerged as a game-changer: fraud detection in financial applications, financial time-series forecasting, predictive and prescriptive analytics, medical image processing, power systems research and recommender systems. The thrust of this review is to outline emerging applications of DL and provide a reference to researchers seeking to use DL in their work for pattern recognition with unparalleled learning capacity and the ability to scale with data.

1. Introduction

Artificial neural networks (ANNs), now one of the most widely-used approaches to computational intelligence, started as an attempt to mimic adaptive biological nervous systems in software and customized hardware [1]. ANNs have been studied for more than 70 years [2] during which time they have waxed and waned in the attention of researchers. Recently they have made a strong resurgence as pattern recognition tools following pioneering work by a number of researchers [3]. It has been demonstrated unequivocally that multilayered artificial neural architectures can learn complex, non-linear functional mappings, given sufficient computational resources and training data. Importantly, unlike more traditional approaches, their results scale with training data. Following these remarkable, significant results in robust pattern recognition, the intellectual neighborhood has seen exponential growth, both in terms of academic and industrial research. Moreover, multilayer ANNs reduce much of the manual work that until now has been needed to set up classical pattern recognizers. They are, in effect, black box systems that can deliver, with minimal human attention, excellent performance in applications that require insights from unstructured, high-dimensional data [4–9]. These facts motivate this review of the topic.

1.1. What is an artificial neural network?

An artificial neural network comprises many interconnected, simple functional units, or *neurons* that act in concert as parallel information-processors, to solve classification or regression problems. That is they can separate the input space (the range of all

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2020.105596.

* Corresponding author.
E-mail addresses: vravi@idrbt.ac.in, rav_padma@yahoo.com (V. Ravi).

https://doi.org/10.1016/j.knosys.2020.105596
0950-7051/

possible values of the inputs) into a discrete number of classes or they can approximate the function (the black box) that maps inputs to outputs. If the network is created by stacking layers of these multiply connected neurons the resulting computational system can:

1. Interact with the surrounding environment by using one layer of neurons to receive information (these units are known to be part of the *input layers* of the neural network)
2. Pass information back-and-forth between layers within the black-box for processing by invoking certain *design goals* and *learning rules* (these units are known to be part of the *hidden layers* of the neural network)
3. Relay processed information out to the surrounding environment via some of its atomic units (these units are known to be part of the *output layers* of the *neural network*).

Within a hidden layer each neuron is connected to the outputs of a subset (or all) of the neurons in the previous layer each multiplied by a number called a *weight*. The neuron computes the sum of the products of those outputs (its inputs) and their corresponding weights. This computation is the dot product between an input vector and weight vector which can be thought of as the projection of one vector onto the other or as a measure of similarity between the two. Assume the input vectors and weights are both n-dimensional and there are m neurons in the layer. Each neuron has its own weight vector, so the output of the layer is an m-dimensional vector computed as the input vector pre-multiplied by an $m \times n$ matrix of weights. That is, the output is an m-dimensional vector that is the linear transformation of an n-dimensional input vector. The output of each neuron is in effect a linear classifier where the weight vector defines a borderline between two classes and where the input vector lies some distance to one side of it or the other. The combined result of all m neurons is an m-dimensional hyperplane that independently classifies the n dimensions of the input into two m-dimensional classes in the output. If the weights are derived via least mean-squared (LMS) estimation from matched pairs of input–output data they form a linear regression, *i.e.* the hyperplane that is closest in the LMS sense to all the outputs given the inputs.

The hyperplane maps new input points to output points that are consistent with the original data, in the sense that some *error function* between the computed outputs and the actual outputs in the training data is minimized. Multiple layers of linear maps, wherein the output of one linear classifier or regression is the input of another, is actually equivalent to a different single linear classifier or regression. This is because the output of k different layers reduces to the multiplication of the inputs by a single $q \times n$ matrix that is the product of the k matrices, one per layer.

To classify inputs non-linearly or to approximate a nonlinear function with a regression, each neuron adds a numerical bias value to the result of its input sum of products (the linear classifier) and passes that through a nonlinear *activation function*. The actual form of the activation function is a design parameter. But they all have the characteristic that they map the real line through a monotonic increasing function that has an inflection point at zero. In a single neuron, the bias effectively shifts the inflection point of the activation function to the value of the bias itself. So the sum of products is mapped through an activation function centered on the bias. Any pair of activation functions so defined are capable of producing a pulse between their inflection points if each one is scaled and one is subtracted from the other. In effect each pair of neurons samples the input space and outputs a specific value for all inputs within the limits of the pulse. Given training data consisting of input–output pairs – input vectors each with a corresponding output vector – the ANN learns an approximation to the function that produced each of the outputs from its corresponding input. That approximation is the partition of the input space into samples that minimizes the error function between the output of the ANN given its training inputs and the training outputs. This is stated mathematically by the *universal approximation theorem* which implies that any functional mapping between input vectors and output vectors can be approximated to with arbitrary accuracy with an ANN provided that it has a sufficient number of neurons in a sufficient number of layers with a specific activation function [10–13]. Fig. 1 shows a perceptron learning model that captures these ideas.

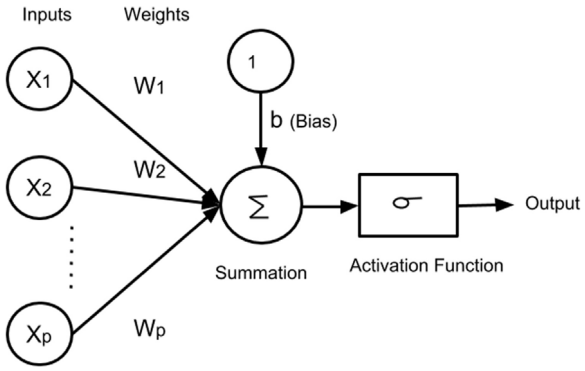

Fig. 1. The perceptron learning model.

Given the dimensions of the input and output vectors, the number of layers, the number of neurons in each layer, the form of the activation function, and an error function, the weights are computed via optimization over input–output pairs to minimize the error function. That way the resulting network is a best approximation of the known input–output data.

1.2. How do these networks learn?

Neural networks are capable of learning — by changing the distribution of weights it is possible to approximate a function representative of the patterns in the input. The key idea is to re-stimulate the black-box using new excitation (data) until a sufficiently well-structured representation is achieved. Each stimulation redistributes the neural weights a little bit — hopefully in the right direction, given the learning algorithm involved is appropriate for use, until the error in approximation w.r.t some well-defined metric is below a practitioner-defined lower bound. Learning then, is the aggregation of a variable length of causal chains of neural computations [14] seeking to approximate a certain pattern recognition task through linear/nonlinear modulation of the activation of the neurons across the architecture. The instances in which chains of implicit linear activation fail to learn the underlying structure, non-linearity aids the modulation process. The term '*deep*' in this context is a direct indicator of the space complexity of the aggregation chain across many *hidden layers* to learn sufficiently detailed representations. Theorists and empiricists alike have contributed to an exponential growth in studies using Deep Neural Networks, although generally speaking, the existing constraints of the field are well-acknowledged [15–17]. Deep learning has grown to be one of the principal components of contemporary research in artificial intelligence in light of its ability to scale with input data and its capacity to generalize across problems with similar underlying feature distributions, which are in stark contrast to the hard-coded, problem-specific pattern recognition architectures of yesteryear (see Fig. 1).

Table 1
Some key advances in neural networks research.

People involved	Contribution
McCulloch & Pitts	ANN models with adjustable weights (1943) [2]
Rosenblatt	The Perceptron Learning Algorithm (1957) [18]
Widrow and Hoff	Adaline (1960), Madaline Rule I (1961) & Madaline Rule II (1988) [19,20]
Minsky & Papert	The XOR Problem (1969) [21]
Werbos (doctoral dissertation)	Backpropagation (1974) [22]
Hopfield	Hopfield Networks (1982) [23]
Rumelhart, Hinton & Williams	Renewed interest in backpropagation: multilayer adaptive backpropagation (1986) [24]
Vapnik, Cortes	Support Vector Networks (1995) [25]
Hochreiter & Schmidhuber	Long Short Term Memory Networks (1997) [26]
LeCunn et. al.	Convolutional Neural Networks (1998) [27]
Hinton & Ruslan	Hierarchical Feature Learning in Deep Neural Networks (2006) [28]

1.3. Why are deep neural networks garnering so much attention now?

Multi-layer neural networks have been around through the better part of the latter half of the previous century. A natural question to ask why deep neural networks have gained the undivided attention of academics and industrialists alike in recent years. There are many factors contributing to this meteoric rise in research funding and volume. Some of these are briefed:

- A surge in the availability of large training data sets with high quality labels
- Advances in parallel computing capabilities and multi-core, multi-threaded implementations
- Niche software platforms such as PyTorch [29], Tensorflow [30], Caffe [31], Chainer [32], Keras [33], BigDL [34] etc. that allow seamless integration of architectures into a GPU computing framework without the complexity of addressing low-level details such as derivatives and environment setup. Table 2 provides a summary of popular Deep Learning Frameworks.
- Better regularization techniques introduced over the years help avoid overfitting as we scale up: techniques like batch normalization, dropout, data augmentation, early stopping etc are highly effective in avoiding overfitting and can single handedly improve model performance with scaling.
- Robust optimization algorithms that produce near-optimal solutions: Algorithms with adaptive learning rates (AdaGrad, RMSProp, Adam, Adaboost), Stochastic Gradient Descent (with standard momentum or Nesterov momentum), Particle Swarm Optimization, Differential Evolution, etc.

1.4. Review methodology

The article, in its present form serves to present a collection of notable work carried out by researchers in and related to the *deep learning* niche. It is by no means exhaustive and limited in its own right to capture the global scheme of proceedings in the ever-evolving complex web of interactions among the deep learning community. While cognizant of the difficulty of achieving the stated goal, we tried to present nonetheless to the reader an overview of pertinent scholarly collections in varied niches in a single article.

The article makes the following contributions from a practitioner's reading perspective:

- It walks through foundations of biomimicry involving artificial neural networks from biological ones, commenting on how neural network architectures learn and why deeper layers of neural units are needed for certain of pattern recognition tasks.
- It talks about how several different deep architectures work, starting from Deep feed-forward networks (DFNNs) and Restricted Boltzmann Machines (RBMs) through Deep Belief Networks (DBNs) and Autoencoders. It also briefly sweeps across Convolutional neural networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs) and some of the more recent deep architectures. This cluster within the article serves as a baseline for further readings or as a refresher for the sections which build on it and follow.
- The article surveys two major computational areas of research in the present day deep learning community that we feel have not been adequately surveyed yet — (a) Multi-agent approaches in automatic architecture generation and learning rule optimization of deep neural networks using swarm intelligence and (b) Testing, troubleshooting and robustness analysis of deep neural architectures which are of prime importance in guaranteeing up-time and ensuring fault-tolerance in mission-critical applications.
- A general survey of developments in certain application modalities is presented. These include:
  Fraud Detection in Financial Services (Section 5.1)
  Financial Time Series Forecasting (Section 5.2)
  Prognostics and Health Monitoring (Section 5.3)
  Medical Imaging (Section 5.4)
  Power Systems (Section 5.5)
  Recommender Systems (Section 5.6)

Fig. 2 captures a high-level hierarchical abstraction of the organization of the review with emphasis on current practices, challenges and future research directions. The content organization is as follows: Section 2 outlines some commonly used deep architectures with a high-level working mechanisms of each, Section 3 talks about the infusion of swarm intelligence techniques within the context of deep learning and Section 4 details diagnostic approaches in assuring fault-tolerant implementations of deep learning systems. Section 5 makes an exploratory survey of several pertinent applications highlighted in the previous paragraph while Section 6 makes a critical analysis of the subtle differences of the review with existing ones. Section 7 makes an in-depth dissection of the general successes and pitfalls of the field as of now before concluding the article.

2. Brief preliminary of deep learning architectures

There are numerous deep architectures available in the literature and growing by the day. A fair comparison of these architectures is a difficult job given different architectures have different advantages based on the application and the characteristics of the data involved. For example in computer vision, Convolutional Neural Networks [27] and in sequence and time series modeling Recurrent Neural Networks [37] are preferred. Deep learning is a fast evolving field and ever so often newer

Table 2
A collection of popular deployment platforms.

Software platform	Purpose
Tensorflow [30]	Software library with high performance numerical computation and support for Machine Learning and Deep Learning architectures compatible to be deployed in CPU, GPU and TPU. url: https://www.tensorflow.org/
Theano [35]	GPU compatible Python library with tight integration to NumPy involves smooth mathematical operations on multidimensional arrays. url: http://deeplearning.net/software/theano/
CNTK [36]	Microsoft Cognitive Toolkit (CNTK) is a Deep Learning Framework describing computations through directed graphs. url: https://www.microsoft.com/en-us/cognitive-toolkit/
Keras [33]	It runs on top of Tensorflow, CNTK or Theano compatible to be deployed in CPU and GPU. url: https://keras.io/
PyTorch [29]	Distributed training and performance evaluation platform integrated with Python supported by major cloud platforms. url: https://pytorch.org/
Caffe [31]	Convolutional Architecture for Fast Feature Embedding (Caffe) is a Deep Learning framework with focus on image classification and segmentation and deployable in both CPU and GPU. url: http://caffe.berkeleyvision.org/
Chainer [32]	Supports CUDA computation and multiple GPU implementation. url: https://chainer.org/
BigDL [34]	Distributed deep learning library for Apache Spark supporting programming languages Scala and Python. url: https://software.intel.com/en-us/articles/bigdl-distributed-deep-learning-on-apache-spark

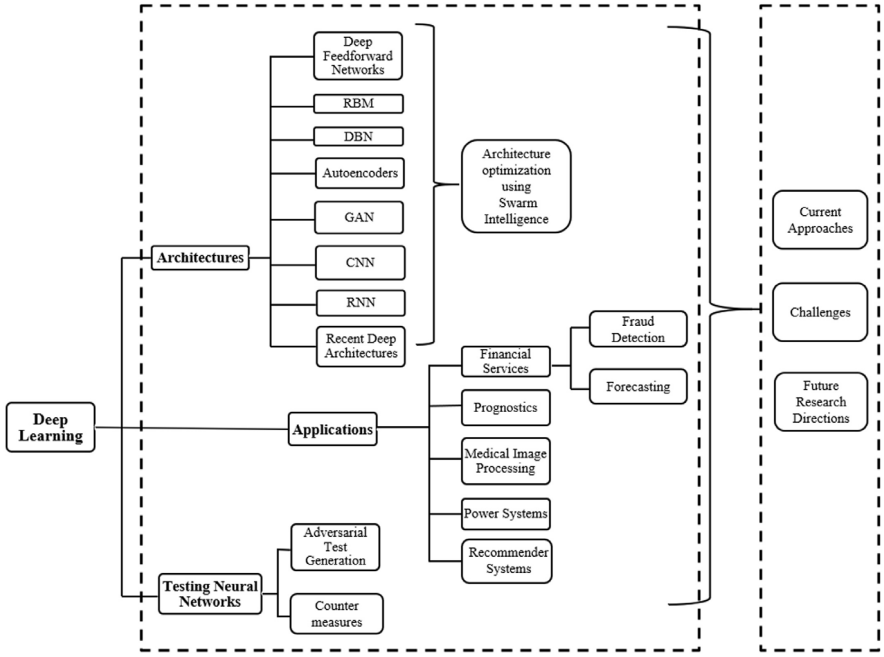

Fig. 2. Organization of the review.

architectures with newer learning algorithms are developed to endure the need to develop human-like efficient machines in different application areas.

2.1. Deep feed-forward networks

Deep Feedforward Neural network, the most basic deep architecture with only the connections between the nodes moves forward. Basically, when a multilayer neural network contains multiple numbers of hidden layers, we call it deep neural network [38]. An example of Deep Feed-Forward Network with n hidden layers is provided in Fig. 3. Multiple hidden layers help in modeling complex nonlinear relation more efficiently compared to the shallow architecture. A complex function can be modeled with less number of computational units compared to a similarly performing shallow network due to the hierarchical learning possible with the multiple levels of nonlinearity [39]. Due to the simplicity of architecture and the training in this model, it is always a popular architecture among researchers and practitioners in almost all the domains of engineering. Backpropagation using gradient descent [40] is the most common learning algorithm used to train this model. The algorithm first initializes the weights randomly, and then the weights are tuned to minimize the error using gradient descent. The learning procedure involves multiple forward and backwards passes consecutively. In forward pass, we forward the input towards the output through multiple hidden layers of non-linearity and ultimately compare the computed output with the actual output of the corresponding input. In the

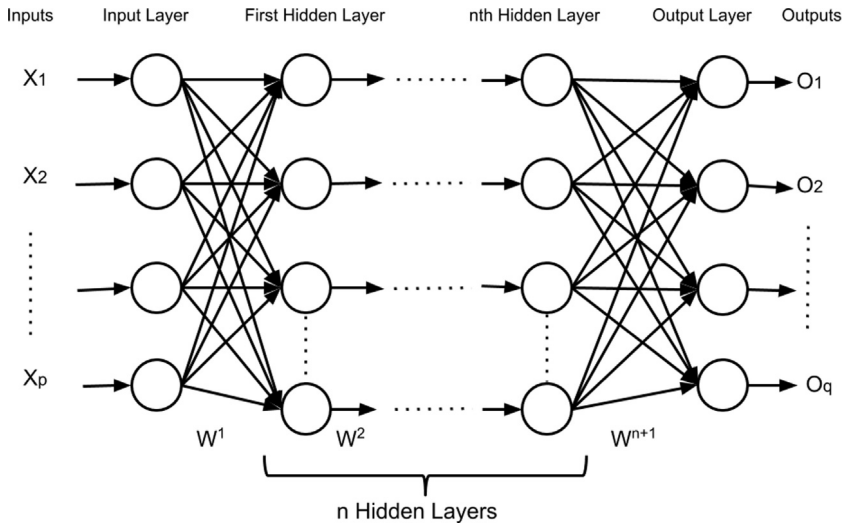

Fig. 3. Deep feed-forward neural network with n hidden layers, p input units and q output units with weights W.

backward pass, the error derivatives with respect to the network parameters are back propagated to adjust the weights in order to minimize the error in the output. The process continues multiple times until we obtained a desired improvement in the model prediction. If X_i is the input and f_i is the nonlinear activation function in layer i, the output of the layer i can be represented by,

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

X_{i+1}, as this becomes input for the next layer. W_i and b_i are the parameters connecting the layer i with the previous layer. In the backward pass, these parameters can be updated with,

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

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

where W_{new} and b_{new} are the updated parameters for W and b respectively, and E is the cost function and η is the learning rate. Depending on the task to be performed like regression or classification, the cost function of the model is decided. For example in regression, root mean square error is common and for classification softmax function.

Many issues like overfitting, trapping in local minima and vanishing gradient problem can arise if a deep neural network is trained naively. Such plaguing issues contributed to sluggish research on neural networks in the late 1990s. However with the advent of unsupervised pre-training approaches in deep neural networks a decade after [28,41], neural network research was revived again to be used for complex tasks like vision and speech. Lately many techniques like L1 and L2 regularization [42], dropout [43], batch normalization [44], a good collection of weight initialization techniques [45–48] and a good set of activation functions [49] have been introduced to combat the longstanding issues in training deep neural networks, with varying degrees of success.

2.2. Restricted Boltzmann machines

Restricted Boltzmann Machines (RBM) [50] can be interpreted as stochastic neural networks. An RBM is a popular deep learning framework due to its ability to learn the input probability distribution in supervised as well as unsupervised manner. It was first introduced by Paul Smolensky in 1986 with the name Harmonium [51]. However, it was popularized by Hinton in 2002 [52] with the advent of improved training algorithms. Subsequently it was applied widely in various tasks like representation learning [53], dimensionality reduction [54], prediction problems [55]. However, deep belief network training using RBMs as building blocks [28] was a very prominent application in the history of RBMs, one that kickstarted the deep learning era along with a handful of other influential breakthroughs discussed later. Recently RBMs are heavily used in the field of collaborative filtering [56] due to their state of the art performance in the Netflix dataset [57].

Restricted Boltzmann Machine is a variation of Boltzmann machine with the restriction in the intra-layer connection between the units, hence the term *restricted*. It is an undirected graphical model containing two layers, visible and hidden, forming a bipartite graph. Different variations of RBMs have been introduced in the literature in terms of improving the learning algorithms, provided the task. Temporal RBMs [58] and conditional RBMs [59] are applied to model multivariate time series data and to generate motion captures. Gated RBMs [60] are used to learn transformation between two input images. Convolutional RBMs [61] are used to understand the time structure of the input time series whereas mean-covariance RBMs [62–64] are applied to represent the covariance structure of the data. There are many other variants such as Recurrent TRBM [65] and factored conditional RBM (fcRBM) [66]. Different types of nodes like Bernoulli, Gaussian [67] etc. are introduced to cope with the characteristics of the data used. However the basic RBM modeling concept was introduced with Bernoulli units. Each node in RBM is a computational unit that processes the input it receives to make stochastic decisions whether to transmit that input or not. An RBM with m visible and n hidden units is provided in Fig. 4.

The joint probability distribution of a standard RBM can be defined with the Gibbs distribution $p(v,h) = \frac{1}{Z} e^{-E(v,h)}$, where energy function E(v,h) can be represented with:

$$E(v,h) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_j v_i - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \tag{4}$$

where m, n are the number of visible and hidden units, v_j, h_j are the states of the visible unit j and hidden unit i, b_j, c_j are the real-valued biases corresponding to the jth visible unit and ith hidden unit respectively, w_{ij} is real-valued weights connecting visible units with hidden units. Z is the normalization constant (sum over all the possible combinations for $e^{-E(v,h)}$) to ensure the probability distributions sums to 1. The restriction made in the intralayer

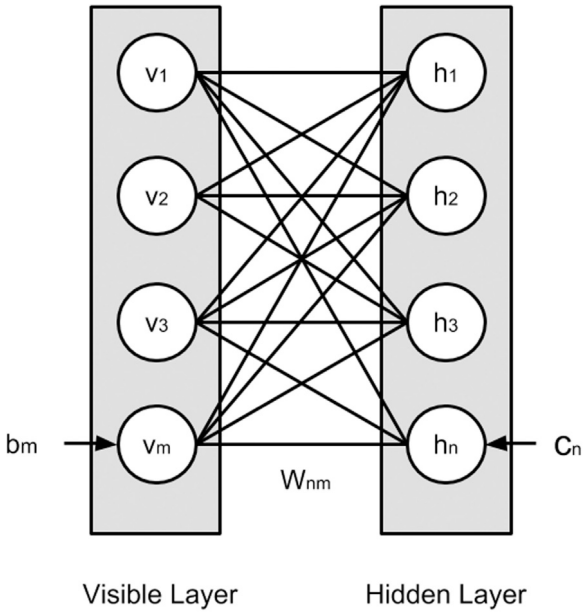

Fig. 4. RBM with m visible units and n hidden units.

connection makes the RBM hidden layer variables independent given the states of the visible layer variables and vice versa. This eases down the complexity of modeling the probability distribution and hence the probability distribution of each variable can be represented by a conditional probability distribution as given below:

$$p(h|v) = \prod_{i=1}^{n} p(h_i|v) \tag{5}$$

$$p(v|h) = \prod_{j=1}^{m} p(v_j|h) \tag{6}$$

RBMs are trained to maximize the expected probability of the training samples. The Contrastive Divergence algorithm proposed by Hinton [52] is popular for training RBMs. The training brings the model to a stable state by minimizing its energy through updates to the parameters of the model. The parameters can be updated using the following equations:

$$\Delta w_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}) \tag{7}$$

$$\Delta b_i = \epsilon(\langle v_i \rangle_{data} - \langle v_i \rangle_{model}) \tag{8}$$

$$\Delta c_j = \epsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}) \tag{9}$$

where ϵ is the learning rate, $\langle . \rangle$ data, $\langle . \rangle$ model are used to represent the expected values of the data and the model.

2.3. Deep belief networks

Deep belief network (DBN) is a generative graphical model composed of multiple layers of latent variables. The latent variables are typically binary, can represent the hidden features present in the input observations. The connection between the top two layers of DBN is undirected like an RBM model, hence a DBN with 1 hidden layer is just an RBM. The other connections in DBN except last are directed graphs towards the input layer. DBN is a generative model, hence to generate a sample from DBN follows a top-down approach. We first draw samples from the RBM on the top layer, this is usually done by Gibbs sampling,

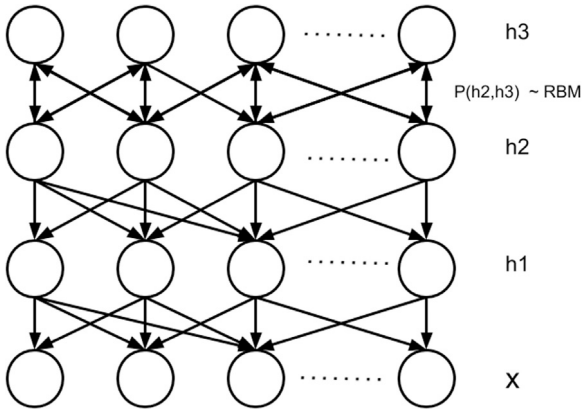

Fig. 5. DBN with input vector **x** with 3 hidden layers.

Then we can perform sampling from the visible units by a simple pass of ancestral sampling in a top-down fashion. A standard DBN model [68] with three hidden layers is shown in Fig. 5.

Inference in DBN is an intractable problem due to the explaining away effect in the latent variable model. However, in 2006 Hinton [28] proposed a fast and efficient way of training DBN by stacking Restricted Boltzmann Machine (RBM) one above the other. The lowest level RBM during training learns the distribution of the input data. The next level of RBM block learns high order correlation between the hidden units of the previous hidden layer by sampling the hidden units. This process is repeated for each hidden layer till the top. A DBN with L numbers of hidden layer models the joint distribution between its visible layer v and the hidden layers h^l, where l = 1, 2, … L as follows:

$$p(v,\ h^1, \ldots,\ h^L) = p(v|\ h^1)(\prod_{l=1}^{L-2} p(h^l|\ h^{l+1}))p(h^{L-1},\ h^L) \tag{10}$$

The log-probability of the training data can be improved by adding layers to the network, which in turn, increases the true representational power of the network [69]. The DBN training proposed in 2006 [28] by Hinton led to the deep learning era of today and revived the neural network. This was the first deep architecture in history trained efficiently. Before this, it was almost infeasible to train deep architectures. Deep architectures built by initializing the weights with DBN outperformed kernel machines, which were prominent in the research landscape of the time. DBNs along with their use as generative models were increasingly applied as discriminative models by appending a discrimination layer at the end and fine-tuning the model using the target labels provided [3]. In most of the applications, this approach of pretraining a deep architecture led to state of the art performance using discriminative models [28,41,54,70] as in recognizing handwritten digits, detecting pedestrians, time series prediction etc. even when the number of labeled data was limited [71]. It garnered immense popularity in acoustic modeling [72] recently as the model could provide up to 20% improvement over other state of the art models, Hidden Markov Models and Gaussian Mixture Models. The approach creates feature detectors hierarchically as "features of features" in pretraining that provide a good set of initialized weights to the discriminative model. The initialized weights are in a region near the optimal weights that can improve both modeling and the convergence speed in fine-tuning [70,73]. DBN has been used as an initialized model in classification in many applications as in phone recognition [62], computer vision [63] etc. where it is used for the training of higher order factorized Boltzmann machines, in speech recognition [74–76]

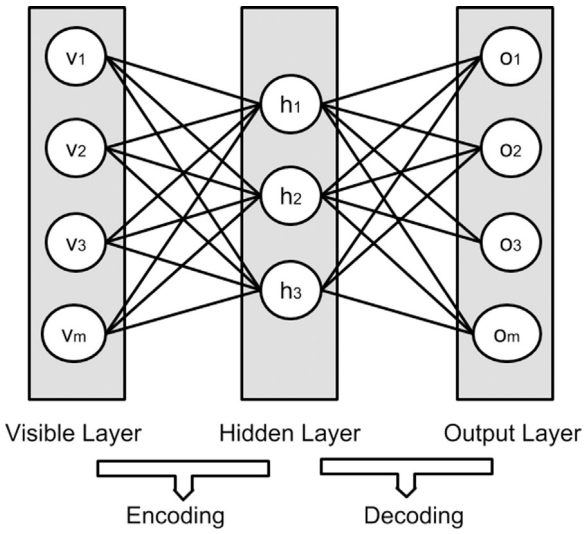

Fig. 6. Autoencoder with 3 neurons in hidden layer.

for pretraining DNN, for pretraining of deep convolutional neural network (CNN) and others [61,77]. The improved performance is due to the ability to learn abstract features using the hidden layer of the network. Some of the work on analysis of the features to understand what is lost and what is captured during its training is demonstrated in [64]. Zhao et al. [78] proposed a parallel computing method for the DBN pre-training and fine-tuning and applied it to a traffic prediction problem. The effectiveness of the method was verified in terms of a decrease in pre-training and fine-tuning times as well as in maintaining the dominant feature learning capacity.

2.4. Autoencoders

An autoencoder is a three-layer neural network, as shown in Fig. 6, which tries to reconstruct its input at its output layer. Hence, the output layer of an autoencoder contains the same number of units as the input layer. The hidden layer typically contains less number of neurons compared to the visible layer and tries to encode or represent the input in a more compact form. It shares the same idea as an RBM, but it typically uses deterministic distributions instead of stochastic units with particular distributions as is the case with RBMs.

Like feedforward neural networks, an autoencoder is typically trained using the backpropagation algorithm. The training consists of two phases: *encoding* and *decoding*. In the *encoding* phase, the model tries to encode the input into some hidden representation using the weight metrics of the lower half layer, and in the *decoding* phase, it tries to reconstruct the same input from the encoding representation using the metrics of the upper half layer. Hence weights in encoding and decoding are forced to be the transpose of each other. The encoding and decoding operation of an autoencoder can be represented by equations below: In encoding phase,

$$y' = f(wx + b) \tag{11}$$

where w, b are the parameters to be tuned, f is the activation function, x is the input vector, and y is the hidden representation. In decoding phase,

$$x' = f(w'y' + c) \tag{12}$$

where w' is the transpose of w, c is the bias to the output layer, x' is the reconstructed input at the output layer. The parameters of the autoencoder can be updated using the following equations:

$$w_{new} = w - \eta \partial E / \partial w \tag{13}$$

$$b_{new} = b - \eta \partial E / \partial b \tag{14}$$

where w_{new} and b_{new} are the updated parameters for w and b respectively at the end of the current iteration and E is the reconstruction error of the input at the output layer.

Autoencoders with multiple hidden layers form deep autoencoders. Similar to deep neural networks, autoencoder training may be difficult due to multiple layers. This can be overcome by training each layer of a deep autoencoder as a simple autoencoder [28,41]. The approach has been successfully applied to encode documents for faster subsequent retrieval [79], image retrieval, efficient speech features [80] etc. As in RBM stacking to form DBN [28] for layerwise pretraining of DNN, autoencoder [41] along with sparse encoding energy-based model [81] are independently developed. Both of these were effectively used to pre-train a deep neural network, much like the DBN. The unsupervised pretraining using autoencoder has been successfully applied in many fields such as image recognition and dimensionality reduction in MNIST [54,80,82], multimodal learning in speech and video images [83,84] and many more. Autoencoders have become immensely popular as generative models in recent years [38,85]. The non-probabilistic and non-generative nature of the conventional autoencoder has been generalized to generative modeling [42,86–89] which can be used to generate meaningful samples from the network.

Several variations of autoencoders are introduced with differing properties and implementations to learn more efficient representation of the data under consideration. One of the popular variation of an autoencoder that is robust to input variations is the denoising autoencoder [42,88,89]. The model can be used for compact representations of input with the number of hidden layers less than the input layer. It can also be used to perform robust modeling of the input distribution with higher number of neurons in the hidden layer. The robustness of the denoising autoencoder is achieved by introducing the dropout trick or by introducing some Gaussian noise to the input data [90,91] or to the hidden layers [92]. This approach helps to improve performance and virtually increases the training set sample size thereby reducing overfitting and making robust representations of the input. Sparse autoencoders [92] are introduced in consideration to allow larger number of hidden units than the visible units in order to make an easier and more efficient representation of the input distribution at the hidden layer. The larger hidden layer represents the input representation by turning on and off its constituent units. Variational autoencoders [85,93] use quite a similar concept as RBMs and learn the stochastic distribution of latent variables instead of deterministic distributions. Transforming autoencoders [94] were proposed as autoencoders with the property of transformation invariance and the encoded features of an autoencoder can effectively reflect this property. The encoder is applied in image recognition [94,95] purposes wherein a “capsule” is used as the building block. A “capsule” is an independent sub-network that extracts local features within a limited window of viewing to check if a feature entity is present with certain probability. Pretraining for CNN using regularized deep autoencoders is actively being researched as of now in computer vision. Robust models of CNN are obtained with denoising autoencoders [87] and sparse autoencoders with pooling and local contrast normalization [96] which provide not only translation-invariant features but also scaling and out-of-plane rotation invariant features.

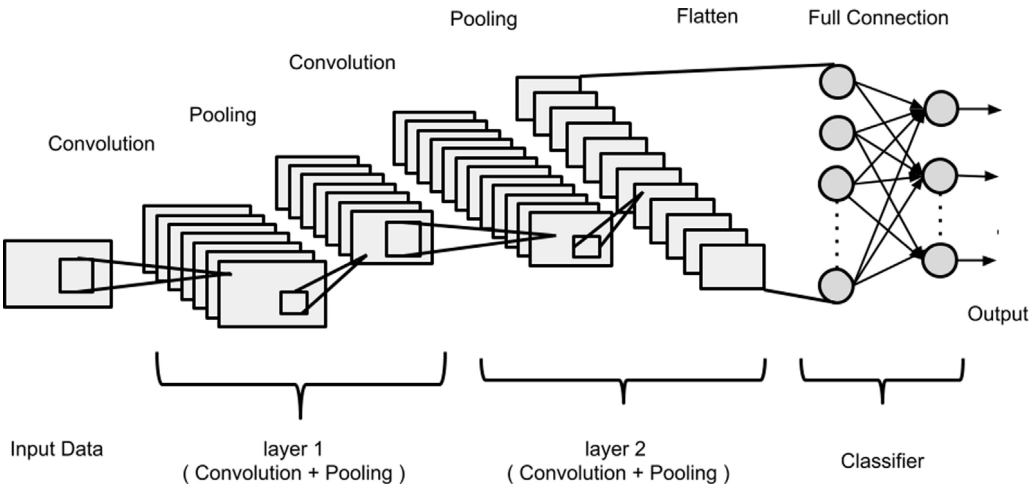

Fig. 7. Convolution and pooling Layers in a CNN.

2.5. Convolutional neural networks

Convolutional Neural Networks (CNN) are a class of neural networks inspired from the human visual system. Although its idea was proposed way back in 1998 by LeCun et al. [97], the deep learning world actually saw it in action when Krizhevsky et al. [98] were able to win the ILSVRC-2012 competition, with the architecture popularly known as AlexNet [98]. This remarkable win started a new era of artificial intelligence and the computation community witnessed the remarkable classification capacities of CNN and its derivatives thereafter. In the years after, many derivative architectures have been proposed and continue to be developed. In many cases, these CNN architectures have been able to outperform human recognition capacities with ease.

The basic architecture of CNN is shown in Fig. 7, which contains multiple convolutions and pooling layers with a fully connected layer at the end. Convolution layers extract important features from the input image in consideration of the spatial relationship between the input pixels whereas pooling layers reduce the dimensionality of the feature map while retaining the feature information [99]. The fully connected layer connects the network with the discriminative layer (output layer), which ultimately provides the desired output. CNNs are particularly useful in extracting image descriptors using latent spatial information. An image has several characteristics like edges, contours, strokes, textures, gradients, orientation, color. A CNN breaks down an image in terms of these type of simple properties and learns them as representations in different layers [100]. Fig. 8 is a good representative of this learning scheme.

CNNs are popular in computer vision tasks such as image detection [102,103], image segmentation [104,105], image classification [106] and image super-resolution reconstruction [107, 108]. Several CNN architectures have been developed considering real-time application requirements while simultaneously meeting high accuracy thresholds. R-CNN (Region-based CNN) [104] and YOLO (You Only Look Once) [109] are examples of such recent architectures. The naive approaches of CNN [110] is computationally very expensive as it considers a massive number of region proposals to locate an object within an image. R-CNN however, is a region-based CNN, that overcomes the limitation of naive CNN by selecting the regions of interest (ROI) with a selective search and limits the proposal regions to 2000 [104]. For the application of R-CNN to real-time processing, the authors later proposed Fast R-CNN [111]. It is a faster approach as unlike the convolution operation on 2000 regions of interest separately on a single image in R-CNN, the convolution operation is performed only once for the whole image. Then the selection search is performed on the generated feature maps to identify region proposals. However, the selection search is still a time-consuming approach and slows down the process of object detection through Fast R-CNN. The time complexity is reduced in Faster R-CNN [112] by replacing the selective search approach with a separate Region Proposal Network (RPN). The R-CNN variations [104,111, 112] discussed above use a two-stage implementation and look at different regions of the image to localize the object within it. It limits the network to fulfill the goal of real-time object detection. YOLO (You Only Look Once) was first proposed by Redmon et al. [109] in 2016 and records very high speed compared to the R-CNN approaches with little compromise in performance. It looks at the object once with a single convolutional neural network and hence understands the generalized representation of the image. However, the algorithm faces spatial constraint in detecting smaller objects. This limitation was considered in Single Shot MultiBox Detector (SSD) [113] which considers different scales of anchor boxes [114] instead of a fixed grid in YOLO. This could effectively handle objects of different sizes, having different resolutions with the ability of real-time inference as in YOLO. Subsequently, different variations of YOLO have been proposed to improve accuracy while keeping the overall pipeline faster. YOLOv2 [115] and YOLOv3 [116] have provided significant improvements in accuracy compared to its previous versions and have also been adapted to detect small objects. Apart from these CNN architectures, there are several variations of existing classical ones such as LeNet [117], AlexNet [98], VGGNet [118], GoogleNet [119], ResNet [120], ZFNet [121] and so on and some of these have been discussed in Section 2.8. The CNN architectures have had a revolutionary impact on and seem to be powering AI-guided vision research for the foreseeable future.

2.6. Recurrent neural networks

Although Hidden Markov Models (HMM) can express time dependencies, they become computationally unfeasible in the process of modeling long term dependencies which RNNs are capable of. A detailed derivation of Recurrent Neural Network from differential equations can be found in [122]. RNNs are form of feed-forward networks spanning adjacent time steps such that at any time instant a node of the network takes the current data input as well as the hidden node values capturing information of previous time steps. Fig. 9 shows a Recurrent Neural Network architecture. During the backpropagation of errors across multiple

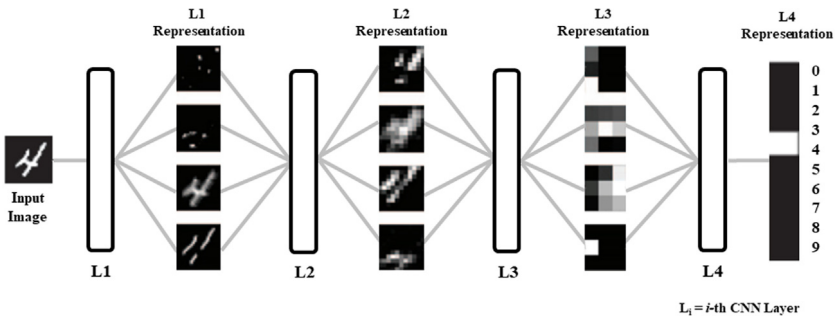

Fig. 8. Representations of an image of handwritten digit learned by CNN [101].

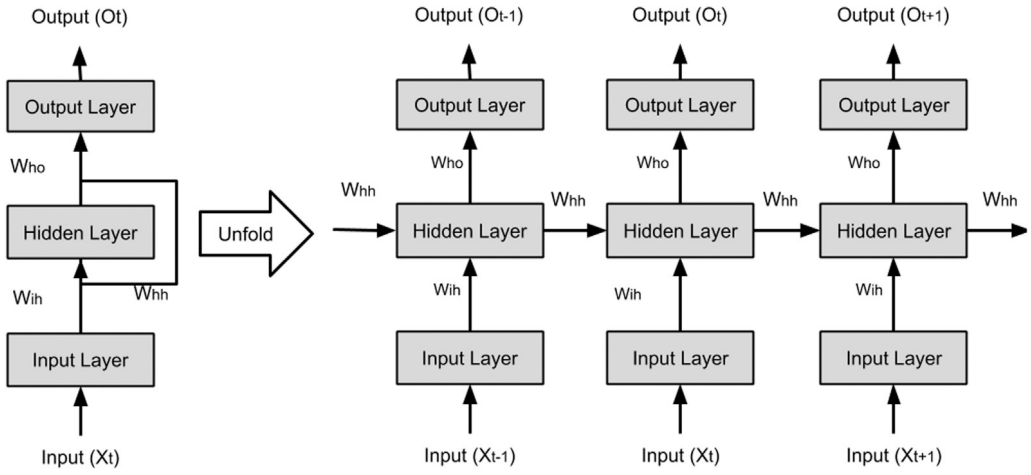

Fig. 9. A recurrent neural network architecture.

timesteps the problem of vanishing and exploding gradients take place which can be avoided by Long Short Term Memory (LSTM) Networks introduced by Hochreiter and Schmidhuber [123]. The amount of information to be retained from previous time steps is controlled by a sigmoid layer known as 'forget' gate whereas the sigmoid activated 'input gate' decides upon the new information to be stored in the cell followed by a hyperbolic tangent activated layer to produce new candidate values which is updated taking forget gate coefficient weighted old state's candidate value. Finally the output is produced controlled by output gate and hyperbolic tangent activated candidate value of the state. Fig. 10 shows a repeating module in LSTM.

LSTM networks with peephole connections [124] updates the three gates using the cell state information. A single update gate instead of forget and input gate is introduced in Gated Recurrent Unit (GRU) [125] merging the hidden and the cell state. In [126] Sak et al. came up with training LSTM RNNs in a distributed way on multicore CPU using asynchronous SGD (Stochastic Gradient Descent) optimization for the purpose of acoustic modeling. They presented a two-layer deep LSTM architecture with each layer having a linear recurrent projection layer with more efficient use of the model parameters. Doetch et al. [127] proposed a LSTM based training framework composed of sequence chunks forming mini batches for training for the purpose of handwriting recognition. With reduction of runtime by a factor of 3 the architecture uses modified gating units with layer specific weights for each gate. Palangi et al. [128] implemented sentence embedding model using LSTM-RNN that sequentially extracts information from each word and embeds in a semantic vector till the end of the sentence to obtain overall semantic representation of the entire sentence. The model with capability of attenuating unimportant words and identifying salient keywords is specifically useful in web document retrieval applications. Pota et al. [129] trained a Bi-LSTM architecture to associate sequence of Part-Of-Speech (POS) tags to a sequence of words, which finds an interesting application in Natural Language Processing. Gao et al. [130] proposed a novel hierarchical attentional module with long short-term memory and multi-layer perceptron to design an end-to-end model for target segregation and localization for visual tracking purposes.

2.7. Generative adversarial networks

Goodfellow et al. [131] introduced a novel framework for Generative Adversarial Nets with simultaneous training of a generative and a discriminative model. The proposed new Generative model bypasses the difficulty of approximation of unmanageable probabilistic measures in Maximum Likelihood Estimation faced previously. The generative model tries to capture the data distribution whereas the discriminative model learns to estimate the probability of a sample either coming from training data or the distribution captured by generative model. If the two above models described by multilayer perceptrons, only backpropagation and dropout algorithms are required to train them. Fig. 11 shows an architecture of a Generative Adversarial Network.

The goal in this process is to train the Generative network in a way to maximize the probability of the discriminative network to make a mistake. A unique solution can be obtained in the function space where the generative model recovers the distribution of training data and the discriminative model results into 50% probability for each sample. This can be viewed as a minmax two player game between these two models as the generative models produce adversarial examples while discriminative model trying to identify them correctly and both try to improve their efficiency until the adversarial examples are indistinguishable from the original ones.

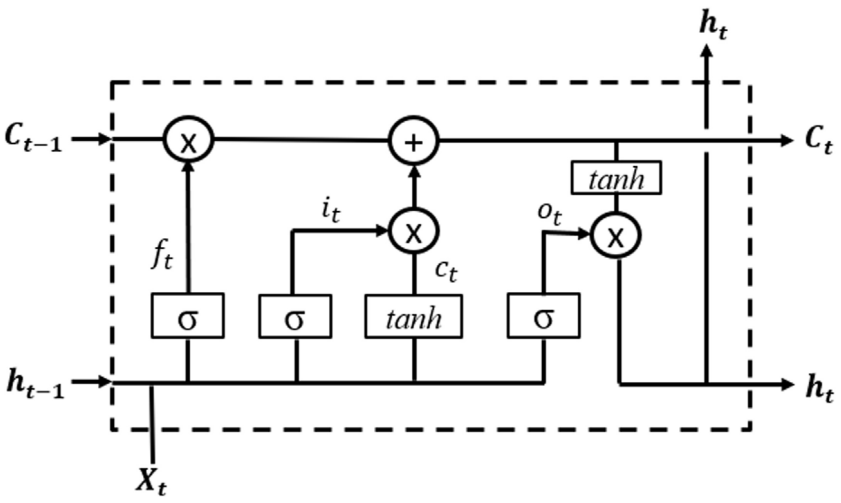

Fig. 10. A repeating module in LSTM.

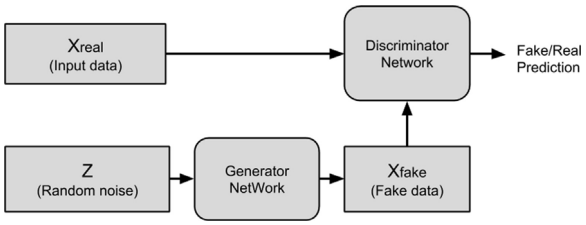

Fig. 11. A generative adversarial network architecture.

In [132], the authors presented training procedures to be applied to GANs focusing on producing visually sensible images. The proposed model was successful in producing MNIST samples visually indistinguishable from the original data and also in learning recognizable features from Imagenet dataset in a semi-supervised way. This work provides insight about appropriate evaluation metric for generative models in GANs and stable semi-supervised training approach. In [133], the authors identified distinct features of GANs from a Turing perspective. The discriminators were allowed to behave as interrogators such as in Turing Test by interacting with data sample generating processes and affirmed the increase in accuracy of the models by verification with two case studies. The first one was about inferring an agent's behavior based on a hidden stochastic process while managing its environment. The second examples talks about active self-discovery exercised by a robot to conclude about its own sensors by controlled movements. Zhou et al. [134] used GAN to generate discriminant faulty data samples in unbalanced data. The major contribution of this paper is in designing a global optimization technique to alternatively update the generator, discriminator and fault diagnosis model.

Wu et al. [135] proposed a 3D Generative Adversarial Network (3DGAN) for three dimensional object generation using volumetric convolutional networks with a mapping from probabilistic space of lower dimension to three dimensional object space so that the 3D object can be sampled or explored without any reference image. As a result high quality 3D objects can be generated employing efficient shape descriptor learnt in an unsupervised manner by the adversarial discriminator. Vondrick et al. [136] came up with video recognition/classification and video generation/prediction model using Generative Adversarial Network (GAN) with separation of foreground from background employing spatio-temporal convolutional architecture. The proposed model is efficient in predicting futuristic versions of static images extracting meaningful features and recognizing actions embedded in the video in a minimally supervised way. Thus, learning scene dynamics from unlabeled videos using adversarial learning is the main objective of the proposed framework.

Another interesting application is generating images from detailed visual descriptions [137]. The authors trained a deep convolutional generative adversarial network (DC-GAN) based on encoded text features through hybrid character-level convolutional recurrent neural network and used manifold interpolation regularizer. The generalizability of the approach was tested by generating images from various objects and changing backgrounds.

2.8. A discussion on recent deep architectures

2.8.1. Some recent CNN architectures

There are several CNN architectures that came up in the last decade. After the success of LeNet [117], Krizhevsky et al. proposed Alexnet [98] in 2012 that accelerated the speed six times given the same accuracy. It used Relu activation function and dropout to control overfitting. Simonyan and Zisserman proposed VGG [118], where they investigated the dependence of image recognition accuracy based on the depth of the CNN architecture. They had kept the kernel size small (3×3) but had pushed the depth up to 16 to 19 layers. As the architecture became larger, the computational overhead increased. Herein came the Googlenet [119] proposed by Szegedy et al. aimed to improve the utilization of computational resources used by the network. In Googlenet, Hebbian principle guided architectural selection was carried out. Later Szegedy et al. proposed Inception-V3 [138] where they investigated the problems of factorizing convolutions, auxiliary classifiers, grid size reductions to train high quality networks with limited computational resources.

In 2016, Y. Bengio proposed ResNet [139] by introducing identity shortcut connections that dealt with the *vanishing gradient problem* as the network goes deeper. Later Xie et al. came up with ResNeXt [140], which is a variant of ResNet that used a cardinality hyperparameter to control the number of independent paths in the architecture for balancing the model capacity. On the other hand, Huang et al. [141] came up with densely connected CNNs that connected all the layers such that an input of a layer contains all previous features extracted and passed it to the next layer. Woo et al. [142] introduced CBAM (Convolutional Block Attention Module) where attention maps are inferred from intermediate feature maps along both spatial and channel dimensions,

which are then used to adaptively refine features. This module is generalizable and can be applied to any CNN architecture. Hu et al. [143] introduced a competitive squeeze-excitation (SE) mechanism for residual networks and re-scaled the value of each channel using both residual and identity mappings which attained state of the art results across datasets such as CIFAR, SVHN [144] and ImageNet.

2.8.2. Some recent RNN architectures

Gregor et al. [145] introduced a novel architecture called Deep Recurrent Attentive Writer (DRAW) combining spatial attention mechanism with sequential variational auto-encoding framework to generate images. The images generated cannot be distinguished from the real images visually based on the results obtained with Street View House Numbers (SVHN) dataset [144]. Kalchbrenner et al. [146] proposed Grid Long Short-Term Memory, where the cells are connected among different layers as well as along the spatiotemporal dimension of the data. Jing et al. [147] added a gating mechanism to unitary RNNs to combine the remembering capability of the unitary RNNs with the ability to forget irrelevant information exhibited by gated RNNs. The proposed method exceeded the performance provided by LSTM, GRU and unitary RNN on various benchmarking problems. Belletti et al. [148] proposed a factorized recurrent network architecture that renders extra memory capacity to capture long-range dependencies.

3. Swarm intelligence in deep learning

The introduction of heuristic and meta-heuristic algorithms in designing complex neural network architectures aimed towards tuning the network parameters to optimize the learning process has brought improvements in the performance of several Deep Learning Frameworks. In order to design the Artificial Neural Networks (ANN) automatically with evolutionary computation a Deep Evolutionary Network Structured Representation (DENSER) was proposed in [149], where the optimal design for the network is achieved by a bi-leveled representation. The outer level deals with the number of layers and their sequence whereas the inner layer optimizes the parameters and hyper parameters associated with each layer defined by a context-free human perceivable grammar. Through automatic design of CNNs the proposed approach performed well on CIFAR-10, CIFAR-100, MNIST and Fashion MNIST dataset. On the other hand, Garro et al. [150] proposed a methodology to automatically design ANN using basic Particle Swarm Optimization (PSO), Second Generation of Particle Swarm Optimization (SGPSO), and a New Model of PSO (NMPSO) to evolve and optimize the synaptic weights, transfer function for each neuron and the architecture itself simultaneously. The ANNs designed in this way were evaluated over eight fitness functions. It aimed towards dimensionality reduction of the input pattern, and was compared to the traditional design architectures using the well known backpropagation and Levenberg–Marquardt algorithms. Das et al. [151] used PSO to optimize the number of layers, neurons, the kind of transfer functions to be involved and the topology of ANN aimed at building channel equalizers that perform better in presence of all noise scenarios.

Wang et al. [152] used Variable-length Particle Swarm Optimization for automatic evolution of deep convolutional neural network architectures for image classification purposes. They proposed novel encoding strategies to encode CNN layers in particle vectors and introduced a disabled layer hiding certain dimensions of the particle vector to have variable-length particles. In addition to this, to speed up the process the authors randomly picked up partial datasets for evaluation. Several PSO-based algorithms and their hybridized versions [153] as well as recent swarm intelligence algorithms such as the QDDS [154] and its chaotic enhancements [155] may be used, among others for automatic generation of deep learning architectures.

Dhariyal and Ravi (2019) [156] proposed a hybrid of Word2Vec, transfer learning (in the form of pre-trained CNN) and differential evolution trained 3-layered neural network (DENN) for sentiment analysis on benchmark movie review datasets available in the web. They reported that it is statistically equivalent to the same architecture except that the DENN is replaced by a probabilistic neural network.

The problem of changing dimensionality of perceived information by each agent in the domain of Deep reinforcement learning (RL) for swarm systems has been solved in [157] using an end-to-end learned mean feature embedding as state information. The research concluded that an end-to-end embedding using neural network features helps to scale up the RL architecture with increasing numbers of agents towards better performing policies as well as ensures fast convergence.

4. Testing neural networks

Software employed in safety critical systems need to be rigorously tested through white-box or black-box testing. In white box testing, the internal structure of the software/program is known and utilized in generating test cases as per the test criteria/requirement. Whereas in black box testing the inputs and outputs of the program are compared as the internal code of the software cannot be accessed. Some of the previous works dealing with generating test cases revealing faulty cases can be found in [158] and in [159] using Principle component analysis. In [160] the authors implemented a black-box testing methodology by feeding randomly generated input test cases to an original version of a real-world test program producing the corresponding outputs, so as the input–output pairs are generated to train a neural network. Then each test case is applied to mutated and faulty version of the test program and compared against the output of the trained ANN to calculate the distance between two outputs indicating whether the faulty program has produced valid or invalid result. Thus ANN is treated as an automated 'oracle' which produces satisfactory results when the training set is comprised of data ensuring good coverage on the whole range of input.

Y. Sun et al. [161] proposed a set of four test coverage criteria drawing inspiration from traditional Modified Condition/Decision Coverage (MC/DC) criteria. They also proposed algorithms for generating test cases for each criterion built upon linear programming. A new test case (an input to Deep Neural Network) is produced by perturbing a given one, where the stated algorithms should encode the test requirement and a fragment of the DNN by fixing the activation pattern obtained from the given input example, and then minimize the difference between the new and the current inputs. The utility of this method lies in bug finding, determining DNN safety statistics, measuring testing accuracy and analysis of DNN internal structure. The paper discusses about sign change, value change and distance change of a neuron pair with two neurons in adjacent layers in the context of their change in activation values in two given test cases. Four covering methods: sign cover, distance–sign cover, sign–value cover and distance–value cover are explained along with test requirement and test criteria which computes the percentage of the neuron pairs that are covered by test cases with respect to the covering method.

For each test requirement an automatic test case generation algorithm is implemented based on Linear Programming (LP). The objective is to find a test input variable, whose value is to be synthesized with LP, with identical activation pattern as a given input. Hence a pair of inputs that satisfy the closeness definition are called adversarial examples if only one of them is correctly

labeled by the DNN. The testing criteria necessitates that (sign or distance) changes of the condition neurons should support the (sign or value) change of every decision neuron. For a pair of neurons with a specified testing criterion, two activation patterns need to be found such that the two patterns together shall exhibit the changes required by the corresponding testing criterion. In the final test suite the inputs matching these patterns will be added. The authors put forward results on 10 DNNs with the Sign–Sign, Distance–Sign, Sign–value and Distance–Value covering methods that show that the test generation algorithms are effective, as they reach high coverage for all covering criteria. Also, the covering methods designed are useful. This is supported by the fact that a significant portion of adversarial examples have been identified. To evaluate the quality of obtained adversarial examples, a distance curve to see how close the adversarial example is to the correct input has been plotted. It is observed that when going deeper into the DNN, it can become harder for the cover of neuron pairs. Under such circumstances, to improve the coverage performance, the use of larger data set when generating test pairs is needed. Interestingly, it seems that most adversarial examples can be found around the middle layers of all DNNs tested. In future the authors propose to find more efficient test case generation algorithms that do not require linear programming.

Katz et al. [162], provided methods for verifying adversarial robustness of neural networks with Reluplex algorithm, to prove, that a small perturbation to a rightly classified input should not result into misclassification. Huang et al. [163], proposed an automated verification framework based on Satisfiability Modulo Theory (SMT) to test the safety of neural network by searching adversarial manipulations through exploration in the space around a given data point. The adversarial examples discovered were used to fine-tune the network.

4.1. Different methods of adversarial test generation

Despite the success of deep learning in various domains, the robustness of the architectures need to be studied before applying neural network architectures in safety critical systems. In this subsection we discuss the kind of malicious attack that can fool or mislead NN to output wrong decisions and ways to overcome them. The work presented by Tuncali et al. [164] deals with generating scenarios leading to unexpected behaviors by introducing perturbations in the testing conditions. For identifying falsification and critical systems behavior for autonomous driving systems, the authors focused on finding glancing counterexamples which refer to the borderline behavior of the system where it is in the verge of failing. They introduced Signal Temporal Logic (STL) formula for the problem in hand which in this case is a combination of predicates over the speed of the target car and distances of all other objects (including cars and pedestrians) and relative positions of them. Then a list of test cases is created and evaluated against STL specification. A covering array spanning all possible combinations of the values the variables can take is generated. To find a glancing behavior, the discrete parameters from the covering array that correspond to the trace that minimize STL conditions for a trace, are used to create test cases either uniformly randomly or by a cost function to guide a search over the continuous variables. Thus, a glancing test case for a trace is obtained. The proposed closed loop architecture behaves in an integrated way along with the controller and Deep Neural Network (DNN) based perception system to search for critical behavior of the vehicle.

In [165] Yuan et al. discuss adversarial falsification problem explaining false positive and false negative attacks, white box attacks where there is complete knowledge about the trained NN model and black box attack where no information of the model can be accessed. With respect to adversarial specificity there are targeted and non-targeted attacks where the class output of the adversarial input is predefined in the first case and arbitrary in the second case. They also discuss about perturbation scope where individual attacks are geared towards generating unique perturbations per input whereas universal attacks generate similar attack for the whole dataset. The perturbation measurement is computed as p-norm distance between actual and adversarial input. The paper discusses various attack methods including L-BFGS attack, Fast Gradient Sign Method (FGSM) by performing update of one step gradient along the direction of the sign of the gradient of every pixel expressed as [166]:

$$\eta = \epsilon sign(\nabla_x J_\theta(x, l)) \tag{15}$$

where ϵ is the magnitude of perturbation η which when added to an input data generates an adversarial data.

FGSM has been extended by Basic Iterative Method (BIM) and Iterative Least-Likely Class Method (ILLC). Moosavi-Dezfooli et al. [167] proposed Deepfool where iterative attack was performed with linear approximation to surpass the nonlinearity in multidimensional cases.

4.2. Countermeasures for adversarial examples

The paper [165] deals with reactive countermeasures such as Adversarial Detecting, Input Reconstruction, and Network Verification and proactive countermeasures such as Network Distillation, Adversarial (Re)training, and Classifier Robustifying. In Network Distillation high temperature softmax activation reduces the sensitivity of the model towards small perturbations. In Adversarial (Re)training adversarial examples are used during training. Adversarial detecting deals with finding the probability of a given input being adversarial or not. In input reconstruction technique a denoising autoencoder is used to transform the adversarial examples to actual data before passing them as input to the prediction module by deep NN. Also, Gaussian Process Hybrid Deep Neural Networks (GPDNNs) are proven to be more robust towards adversarial inputs.

There are also ensembling defense strategies to counter multi-faceted adversarial examples. But the defense strategies discussed here are mostly applicable to computer vision tasks, whereas the need of the day is to generate real time adversarial input detection and take measures for safety critical systems.

In [168] Rouhani et al. proposed an online defense framework DeepFense against adversarial deep learning. They formulated it as an unsupervised optimization problem by minimizing the less observed spaces in the latent feature hyperspace spanned by a Deep Learning network and was able to decrease the risk of integrated attacks. With integrated design of algorithms for software and hardware the proposed framework aims to maximize model reliability.

It is necessary to build robust countermeasures to be used for different types of adversarial scenarios to provide a reliable infrastructure as none of the countermeasures can be universally applicable to all sorts of adversaries. A detailed list of specific attack generation and corresponding countermeasures can be found in [169] (see Table 3).

5. Applications

5.1. Fraud detection in financial services

Fraud detection is an interesting problem in that it can be formulated in an unsupervised, a supervised and a one-class classification setting. In unsupervised learning category, class labels

Table 3
Distribution of surveyed articles by application areas.

Application area	Authors
Fraud Detection in Financial Services	Pumsirirat et al. [170], Schreyer et al. [171], Wang et al. [172], Gangwar and Ravi [173], Zheng et al. [174], Dong et al. [175], Gomez et al. [176], Rymantubb et al. [177], Fiore et al. [178], Dhariyal and Ravi [156]
Financial Time Series Forecasting	Cavalcante et al. [179], Li et al. [180], Fama et al. [181], Lu et al. [182], Tk & Verner [183], Pandey et al. [184], Lasfer et al. [185], Gudelek et al. [186], Fischer & Krauss [187], Santos Pinheiro & Dras [188], Bao et al. [189], Hossain et al. [190], Calvez and Cliff [191]
Prognostics and Health Monitoring	Basak et al. [192], Tamilselvan & Wang [193], Schroeder et al. [194], Wang et al. [195], Kuremoto et al. [196], Qiu et al. [197], Gugulothu et al. [198], Filonov et al. [199], Botezatu et al. [200], Fei et al. [201], Zhang et al.[202], Ma et al.[203]
Medical Image Processing	Suk, Lee & Shen [204], van Tulder & de Bruijne [205], Brosch & Tam [206], Esteva et al. [207], Rajaraman et. al. [208], Kang et al. [209], Hwang & Kim [210], Andermatt et al. [211], Cheng et al. [212], Miao et al. [213], Oktay et al. [214], Golkov et al. [215], Litjens et al. [216], Yildirim, Tan and Acharya [217], Gangwar and Ravi [218], Yildirim et al. [219], Raghavendra et al. [220], Talo et al. [221], Baloglu et al. [222], Talo et al. [223]
Power Systems	Vankayala & Rao [224], Chow et al. [225], Guo et al. [226], Bourguet & Antsaklis [227], Bunn & Farmer [228], Hippert et al. [229], Kuster et al. [230], Aggarwal & Song [231], Zhai [232], Park et al. [233], Mocanu et al. [234], Chen et al. [235], Bouktif et al. [236], He et al. [237], Li et al. [238], Cecati et al. [239], Dedinec et al. [240], Rahman et al. [241], Kong et al. [242], Dong et al. [243], Kalogirou et al. [244], Wang et al. [245], Das et al. [246], Dabra et al. [247], Liu et al. [248], Jang et al. [249], Gensler et al. [250], Abdel-Nasser et al. [251], Manwell et al. [252], Marugán et al. [253], Wu et al. [254], Wang et al. [255], Wang et al. [256], Qureshi et al. [257]
Recommender Systems	Marz & Warren [258], Adomavicius & Tuzhilin [259], Bokde, Girase & Mukhopadhyay [260], Sedhain et al. [261], Salakhutdinov [56], Wu et al. [262], Wang et al. [263], Georgiev et al. [264], Liu et al. [265], Hongliang and Xiaona [266], Wang and Wang [267], Van den Oord, Dieleman and Schrauwen [268], Zheng et al. [269], Kim et al. [270], He et al. [271], Tay et al. [272]

either unknown or are assumed to be unknown and clustering techniques are employed to figure out (i) distinct clusters containing fraudulent samples or (ii) far off fraudulent samples that do not belong to any cluster, where all clusters contained genuine samples, in which case, it is treated as an outlier detection problem. In supervised learning category, class labels are known and a binary classifier is built in order to classify fraudulent samples. Examples of these techniques include logistic regression, Naive Bayes, supervised neural networks, decision tree, support vector machine, fuzzy rule based classifier, rough set based classifier etc. Finally, in the one-class classification category, only samples of genuine class available or fraud samples are not considered for training even if available. These are called one-class classifiers. Examples include one-class support vector machine (aka Support vector data description or SVDD), auto association neural networks (aka auto encoders). In this category, models are trained on the genuine class data and are tested on the fraud class. Literature abounds with many studies involving traditional neural networks with various architectures to deal with the above mentioned three categories. Having said that fraud (including cyber fraud) detection is increasingly becoming menacing and fraudsters always appear to be few notches ahead of organizations in terms of finding new loopholes in the system and circumventing them effortlessly. On the other hand, organizations make huge investments in money, time and resources to predict fraud in near real-time, if not real time and try to mitigate the consequences of fraud. Financial fraud manifests itself in various areas such as banking, insurance and investments (stock markets). It can be both offline as well as online. Online fraud includes credit/debit card fraud, transaction fraud, cyber fraud involving security, while offline fraud includes accounting fraud, forgeries etc.

Deep learning algorithms proliferated during the last five years having found immense applications in many fields, where the traditional neural networks were applied with great success. Fraud detection one of them. In what follows, we review the works that employed deep learning for fraud detection and appeared in refereed international journals and one article is from arXive repository. Papers published in International conferences are excluded.

Pumsirirat (2018) [170] proposed an unsupervised deep auto encoder (AE) based on restricted Boltzmann machine (RBM) in order to detect novel frauds because fraudsters always try to be innovative in their modus operandi so that they are not caught while perpetrating the fraud. He employed backpropagation trained deep Auto-encoder based on RBM that can reconstruct normal transactions to find anomalies from normal patterns. He used the Tensorflow library from Google to implement AE, RBM, and H2O by using deep learning. The results show the mean squared error, root mean squared error, and area under curve.

Schreyer (2017) [171] observed the disadvantage of business and experiential-knowledge driven rules in failing to generalize well beyond the known scenarios in large scale accounting frauds. Therefore, he proposed a deep auto encoder for this purpose and tested it effectiveness on two real world datasets. Chartered accountants appreciated the power of the deep auto encoder in predicting the anomalous accounting entries.

Automobile insurance fraud has traditionally been predicted by considering only structured data and textual date present in the claims was never analyzed. But, Wang and Xu (2018) [172] proposed a novel method, wherein Latent Dirichlet Allocation (LDA) was first used to extract the text features hidden in the text descriptions of the accidents appearing in the claims, and then along with the traditional numeric features as input data deep neural networks are trained. Based on the real-world insurance fraud dataset, they concluded their hybrid approach outperformed random forests and support vector machine.

Telecom fraud has assumed large proportions and its impact can be seen in services involving mobile banking. Zheng et al. (2018) [174] proposed a novel generative adversarial network (GAN) based model to compute probability of fraud for each large transfer so that the bank can prevent potential frauds if the probability exceeds a threshold. The model uses a deep denoising autoencoder to learn the complex probabilistic relationship among the input features, and employs adversarial training to accurately discriminate between positive samples and negative samples in a data. They concluded that the model outperformed traditional classifiers and using it two commercial banks have reduced losses of about 10 million RMB in twelve weeks thereby significantly improving their reputation.

In today's word-of-mouth marketing, online reviews posted by customers critically influence buyers' purchase decisions more

than before. However, fraud can be perpetrated in these reviews too by posting fake and meaningless reviews, which cannot reflect customers'/users' genuine purchase experience and opinions. They pose great challenges for users to make right choices. Therefore, it is desirable to build a fraud detection model to identify and weed out fake reviews. Dong et al. (2018) [175] present an autoencoder and random forest, where a stochastic decision tree model fine tunes the parameters. Extensive experiments were conducted on a large Amazon review dataset.

Gomez et al. (2018) [176] presented a neural network based system for fraud detection in banking. They analyzed a real world dataset, and proposed an end-to-end solution from the practitioner's perspective, especially focusing on issues such as data imbalances, data processing and cost metric evaluation. They reported their proposed solution performed comparably with state-of-the-art solutions.

Ryman-Tubb et al. (2018) [177] observed that payment card fraud has dented economies to the tune of USD 416bn in 2017. This fraud is perpetrated primarily to finance terrorism, arms and drug crime. Until recently the patterns of fraud and the criminals modus operandi has remained unsophisticated. However, smart phones, mobile payments, cloud computing and contactless payments have emerged almost simultaneously with large-scale data breaches. This made the extant methods less effective. They surveyed extant methods using transactional volumes in 2017. This benchmark will show that only eight traditional methods have a practical performance to be deployed in industry. Further, they suggested that a cognitive computing approach and deep learning are promising research directions.

Fiore et al. (2019) [178] observed that data imbalance is a crucial issue in payment card fraud detection and that oversampling has some drawbacks. They proposed Generative Adversarial Networks (GAN) for oversampling, where they trained a GAN to output mimicked minority class examples, which were then merged with training data into an augmented training set so that the effectiveness of a classifier can be improved. They concluded that a classifier trained on the augmented set outperformed the same classifier trained on the original data, especially as far the sensitivity is concerned, resulting in an effective fraud detection mechanism.

Taking cue from Fiore et al. (2019) [178], Gangwar and Ravi (2019) [173] proposed GAN and Wasserstien GAN, separately, for balancing the minority (class of fraudulent transactions) class in credit card fraud detection. For the classification purpose, they employed logistic regression and support vector machine. They obtained highest precision and least false positive count compared to the state-of-the-art methods.

In summary, as far as fraud detection is concerned, some progress is made in the application of a few deep learning architectures. However, there is immense potential to contribute to this field especially, the application of Resnet, gated recurrent unit, capsule network etc to detect frauds including cyber frauds.

5.2. Financial time series forecasting

Advances in technology and break through in deep learning models have seen an increase in intelligent automated trading and decision support systems in Financial markets, especially in the stock and foreign exchange (FOREX) markets. However, time series problems are difficult to predict especially financial time series [179]. On the other hand, NN and deep learning models have shown great success in forecasting financial time series [180] despite the contradictory report by efficient market hypothesis (EMH) [181], that the FOREX and stock market follows a random walk and any profit made is by chance. This can be attributed to the ability of NN to self-adapt to any nonlinear data set without any statically assumption and prior knowledge of the data set [182].

Deep leaning algorithms have used both fundamental and technical analysis data, which is the two most commonly used techniques for financial time series forecasting, to trained and build deep leaning models [179]. Fundamental analysis is the use or mining of textual information like financial news, company financial reports and other economic factors like government policies, to predict price movement. Technical analysis on the other hand, is the analysis of historical data of the stock and FOREX market.

Deep Learning NN (DLNN) or Multilayer Feed forward NN (MFF) is the most used algorithms for financial markets [183]. According to the experimental analysis done by Pandey et al. [184], showed that MFF with Bayesian learning performed better than MFF learning with back propagation for the FOREX market. The network architectures for both Bayesian learning and back-propagation learning MFF were (7-6-5-1), (7-4-2-1) and (5-10-1). The performance of Bayesian learning was attribute to its ability to avoid over-fitting. Bayesian learning use the given data to infer the posterior distribution of the parameters.

Deep neural networks or machine learning models are considered as a black box, because the internal workings is not fully understood. The performance of DNN is highly influence by the its parameters for a particular domain. Lasfer et al. [185] performed an analysis on the influence of parameter (like the number of neurons, learning rate, activation function etc.) on stock price forecasting. The authors work showed that a larger NN produces a better result than a smaller NN. However, the effect of the activation function on a large NN is lesser.

Although CNN is well known for its stripes in image recognition and less application in the Financial markets, CNN have also shown good performance in forecasting the stock market. As indicated by [185], the deeper the network the more NN can generalize to produce good results. However, the more the layers of NN, it is more likely to overfit a given data set. CNN on the other hand, with its techniques of convolution, pooling and drop out mechanism reduces the tendency of overfitting [186].

In order to apply CNN for the Financial market, the input data need to be transformed or adapted for CNN. With the help of a sliding window, Gudelek et al. [186] used 2D images generated by taking snapshots of the stock time series data and then fed it into 2D-CNN to perform daily predictions and classification of trends (whether downwards or upwards). The 2D-CNN architecture was made up of 784 neurons input layers, two convolution layers of 3×3 32 and 64 filters respectively, 64 4×4 max pooling, and 25 percent of neurons were selected with the help of dropout as inputs for a fully connected NN 128 neurons. Dropout was used to reduce over-fitting [273]. Finally, 50 percent of neurons from the 128 fully connected NN were selected as input for a 64 fully connected NN. An adaptive learning rate method (ADADELTA) optimizer [274] was used to optimize the weights of the CNN. The model was able to get 72 percent accuracy on 17 exchange traded fund data set. The model was not compared against other NN architecture.

Fisher and Krauss [187] adapted LSTM for stock prediction and compared its performance with memory-free based algorithms like random forest, logistic regression classifier and deep neural network. LSTM performed better than other algorithms, random forest however, outperformed LSTM during the financial crisis in 2008.

EMH [181] holds the view that financial news which affects the price movement are in cooperated into the price immediately or gradual. Therefore, any investor that can first analyze the news and make a good trading strategy can profit. Based on this view and the rise of big data, there has been an upward trend in

sentiment analysis and text mining research which utilizes blogs, financial news social media to forecast the stock or FOREX market [179]. Santos et al. [188] explored the impact of news articles on company stock prices by implementing a LSTM neural network pre-trained by a character level language model to predict the changes in prices of a company for both inter day and intraday trading. The authors used an LSTM with a single layer made up of 1024 units which receives a 256 units of embedded character as inputs. The updated weights of the LSTM and the character embeddings were stored and used as the initial weights and inputs for a two layered DNN for classification. The weights of the DNN were updated using stochastic gradient descent (SGD). The final output of the two layer DNN was the input for a fully connected layer with 512 units, which used Leaky Relu activation. The results showed that, CNN with word wise based model outperformed other models. However, LSTM character level-based model performed better than RNN base models and also has less architectural complexity than other algorithms.

Moreover, there has been hybrid architectures to combine the strengths or more than one deep leaning models to forecast financial time series. Bao et al. [189] combined wavelet transform, stacked autoencoders and LSTM for stock price prediction. The stack autoencoders had 10 hidden layers with the depth was set to 5. Input variables were 18 to 28. The hidden layers of the LSTM was also set to 5, while delays was set to 4. Back-propagation algorithm was used to train both the autoencoders and LSTM algorithms. All parameter selected were not based on any rule of thumb, but based on try and error. The output of one network or model was fed into the next model as input. The hybrid model perfumed better than LSTM and RNN (which were standalone). Hossain et al. [190], also created a hybrid model by combining LSTM and Gated recurrent unit (GRU) to predict S&P 500 stock price. The model was compared against standalone models like LSTM and GRU with different architectural layers. The hybrid model outperformed all other algorithms.

Calvez and Cliff [191] introduced a new approach on how to trade on the stock market with DLNN models, which observe and learn the behaviors of traders. The author used a limit-order-book (LOB) and quotes made by successful traders (both automated and humans) as input data. The DLNN model was able to learn and outperform both human traders and automated traders. This approach of learning may be a breakthrough for intelligent automated trading for Financial markets.

5.3. Prognostics and health management

The service reliability of the ever-encompassing cyber-physical systems around us has started to garner the undivided attention of the prognostics community in recent years. Factors such as revenue loss, system downtime, failure in mission-critical deployments and market competitive index are emergent motivations behind making accurate predictions about the State-of-Health (SoH) and Remaining Useful Life (RUL) of components and systems. Industry niches such as manufacturing, electronics, automotive, defense and aerospace are increasingly becoming reliant on expert diagnosis of system health and smart recommender systems for maximizing system uptime and adaptive scheduling of maintenance. Given the surge in sensor influx, if there exists sufficient structured information in historical or transient data, accurate models describing the system evolution may be proposed. The general idea is that in such approaches, there is a point in the operational cycle of a component beyond which it no longer delivers optimum performance. In this regard, the most widely used metric for determining the critical operational cycle is termed as the Remaining Useful Life (RUL), which is a measure of the time from measurement to the critical cycle beyond which sub-optimal performance is anticipated. Prognostic approaches may be divided into three categorizations: (a) Model-driven (b) Data-driven (c) Hybrid i.e. any combination of (a) and (b). The last three decades have seen extensive usage of model-driven approaches with Gaussian Processes and Sequential Monte-Carlo (SMC) methods which continue to be popular in capturing patterns in relatively simpler sensor data streams. However, one shortcoming of model driven approaches used till date happens to be their dependence on physical evolution equations recommended by an expert with problem-specific domain knowledge. For model-driven approaches to continue to perform as well when the problem complexity scales, the prior distribution (physical equations) needs to continue to capture the embedded causalities in the data accurately. However, it has been the observation that as sensor data scales, the ability of model-driven approaches to learn the inherent structures in the data has lagged. This is of course due to the use of simplistic priors and updates which are unable to capture the complex functional relationships from the high dimensional input data. With the introduction of self-regulated learning paradigms such as Deep Learning, this problem of learning the structure in sensor data was mitigated to a large extent because it was no longer necessary for an expert to hand-design the physical evolution scheme of the system. With the recent advancements in parallel computational capabilities, techniques leveraging the volume of available data have begun to shine. One key issue to keep in mind is that the performance of data-driven approaches are only as good as the labeled data available for training. While the surplus of sensor data may act as a motivation for choosing such approaches, it is critical that the precursor to the supervised part of learning, i.e. data labeling is accurate. This often requires laborious and time-consuming efforts and is not guaranteed to result in the generation of near-accurate ground truth. However, when adequate precaution is in place and strategic implementation facilitating optimal learning is achieved, it is possible to deliver customized solutions to complex prediction problems with an accuracy unmatched by simpler, model-driven approaches. Therein lies the holy grail of deep learning: the ability to scale learning with training data.

Fig. 12 shows a generalized prognostic framework that starts with collection of sensor data to gather some information about the underlying system including instances of device performances or features under both normal and degraded operating conditions. The relevant information from the measurements are extracted by feature selection. Then a suitable prognostic approach is applied. Model driven approaches have a physical model describing the behavior of the damage or degradation state and combine the model with measured data to identify model parameters. The model parameters which have an effect on model behavior are often unknown and need to be identified as a part of the prognostic process. In data driven approaches we do not have a physical model. Through observing lots of data instances we can apply Machine Learning/Deep Learning approaches to learn the pattern of the degradation model. If we have information about the physical law governing the system dynamics as well as large volume of data to train on, hybrid approaches can be applied. The goal is to predict the Remaining Useful Life (RUL) of a system to predict the remaining amount of time the system should be in active state before it crosses a threshold beyond which it fails. This requires prognosticating future states of a system in order to ensure proper maintenance. In this case maintaining a balance between false positives and negatives in identifying faulty states plays a crucial role under real world industrial constraints and accuracy measures such as precision and recall must be validated before deployment.

Classical approaches on device health prediction are generally model-driven. While B. Schroeder et al. [194] showed the failure distribution of hard disks can be approximated by Gamma

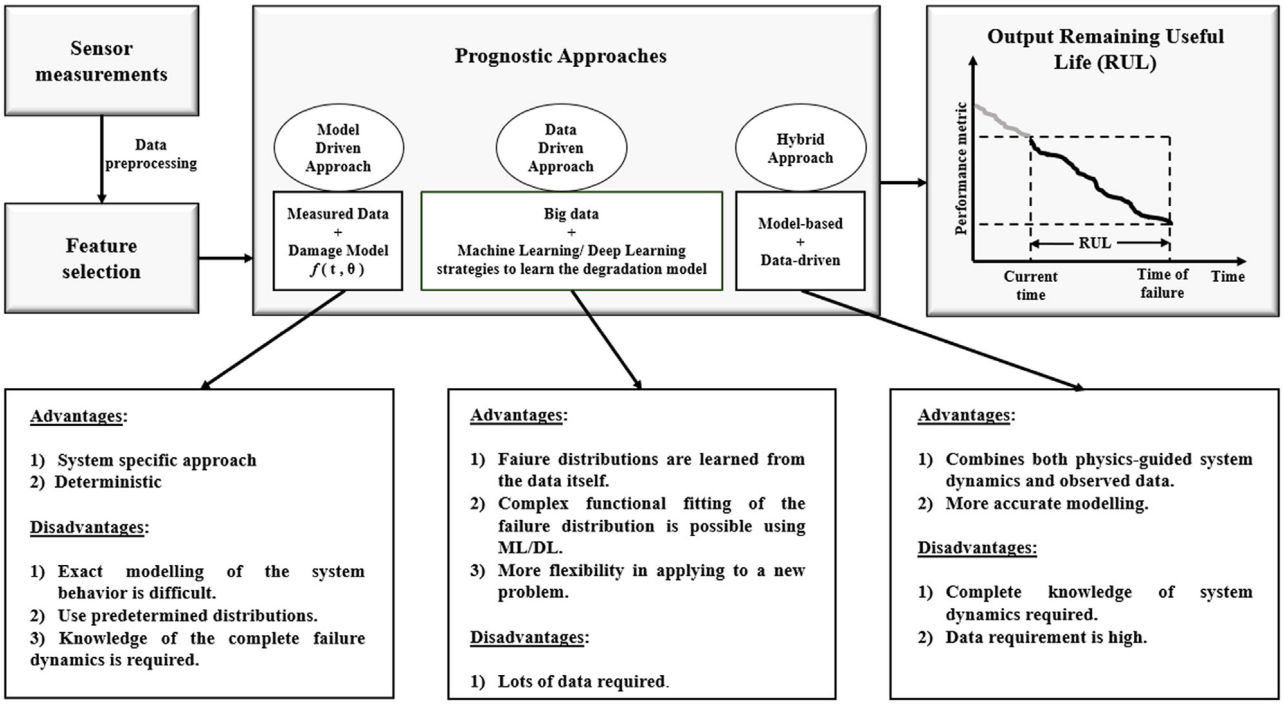

Fig. 12. An illustration of a generalized prognostic framework.

and Weibull distributions, Wang et al. [195] contended this fact and stated that the distribution of time-between-failure is hard to model using a well-known distribution. The recent works on device health forecasting are more focused on data-driven approaches using deep learning techniques. Basak et al. [275] carried out Remaining Useful Life (RUL) prediction of hard disks along with discussions on effective feature normalization strategies on the Backblaze hard disk data [276] using Long Short Term Memory (LSTM) networks. They reported an average precision of 0.84 and recall of 0.72 in identifying whether the disks are going to fail within the next ten days. Deep Belief Network (DBN) based multisensor health diagnosis methodology has been proposed in [193] by Tamilselvan and Wang and employed in aircraft engine and electric power transformer health diagnosis to show the effectiveness of the approach.

Qiu et al. [197] proposed an early warning model where feature extraction through DNN with hidden state analysis of Hidden Markov Model (HMM) is carried out for health maintenance of equipment chain in gas pipeline. The accuracy using their proposed DNN-HMM model on the complete test set was reported as 90.16% and the false alarm rate was 5.07%. Gugulothu et al. [198] proposed a forecasting scheme using a Recurrent Neural Network (RNN) model to generate embeddings which capture the trend of multivariate time series data which are supposed to be disparate for healthy and unhealthy devices. The authors provided various performance metrics indicating their efficiency in identifying RUL for two datasets: engine dataset and pump dataset. The idea of using RNNs to capture intricate dependencies among various time cycles of sensor observations is also emphasized in [199] for prognostic applications.

In various real-world applications, prognosis of the future states of any large scale dynamical system helps to take decisions in advance that are critical for overall system performance. A transportation network is one such dynamically changing network. Prognosticating future traffic states is reliant upon identifying congestions in the network in advance. This can be done with both model-driven and data-driven approaches. The complex behavior of traffic networks is difficult to represent using model-driven approaches as all the modalities of the system cannot be explained by predetermined mathematical distributions [203]. Also the parameter estimation of the model largely depends on the specific scenarios making them mostly infeasible to generalize to any other set of problems. Fei et al. [201] modeled traffic flow in a way such that some of the model parameters include variables specific to a certain type of road segment and not applicable to a different design of road segment/network. They reported an average absolute error of 1.72 km/h in predicting propagation speed using model-driven approach. However while applying it to a new road network, reorientation of the model parameters and re-estimation of their optimal values are required thus making the model not completely generalizable.

On the other hand, Zhang et al. [202] used data-driven techniques to analyze traffic congestion setups. They created a dataset named Seattle Area Traffic Congestion Status (SATCS) and applied a deep auto-encoder based neural network to learn the temporal correlations in order to predict traffic congestion. This resulted in a minimum weighted MSE (wMSE) of 0.0579 for a ten-minute prediction horizon. Ma et al. [203] used conditional Restricted Boltzmann Machine and Recurrent Neural Networks to predict traffic congestions with an average prediction accuracy of 88.2%. In recent years a significant thrust within the community has been on developing data-driven predictive and prescriptive analysis frameworks as sensor data scales and it becomes increasingly infeasible to capture latent dependencies using traditional approaches.

5.4. Medical image processing

Deep learning techniques have pervaded the entire discipline of medical image processing and the number of studies highlighting its application in canonical tasks such as image classification, detection, enhancement, image generation, registration and segmentation have been on a sharp rise. A recent survey by Litjens et al. [216] presents a collective picture of the prevalence and applications of deep learning models within the community as does a fairly rigorous treatise of the same by Shen et al. [277]. A

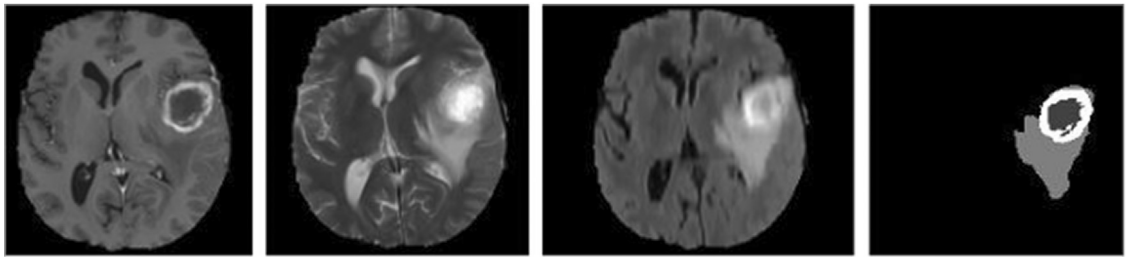

Fig. 13. Brain tumor segmentation [278] . From left: brain images with different MRI modalities. Rightmost figure is the segmented image; bright signal is active region, dark signal is necrotic core and medium level signal is edema. Images are generated by using BRATS 2013 data [279].

concise overview of recent work in some of these canonical tasks follows.

The purpose of medical image classification problems is to aid in identifying the presence of abnormalities in images acquired through medical examinations. Over the last few years various neural network architectures have been used with such motive, including stacked auto-encoders applied for the diagnosis of Alzheimer's disease and mild cognitive impairment, exploiting the latent non-linear complicated relations among various features [204], Restricted Boltzmann Machines applied to Lung Computed Tomography (CT) analysis combining generative as well as discriminative learning techniques [205], Deep Belief Networks trained on three dimensional medical images [206] etc. In recent years, the trend of using Convolutional Neural Networks has been increasingly observed across detection [217,219,220] and classification [221–223] tasks. In 2017, Esteva et al. [207] used and fine-tuned the Inception v3 [280] model to classify clinical images pertaining to skin cancer examinations into benign and malignant variants. Validation of experiments were carried out by testing model performance against a good number of dermatologists. In 2018, Rajaraman et al. [208] used specialized CNN architectures like ResNet for detecting malarial parasites in thin blood smear images. Kang et al. [209] improved the performance of 2D CNN by using a 3D multi-view CNN for lung nodule classification using spatial contextual information with the help of 3D Inception-ResNet architecture. In 2019, Talo et al. [223] classified MR brain images into normal and abnormal categories using a Resnet-34 pre-trained CNN and deep transfer learning techniques. The proposed model yielded a 5-fold automatic classification accuracy of 100% when subjected to 613 MR images.

Object/lesion detection aims to identify different parts/lesions in an image. Although object classification and object detection are quite similar to each other but the challenges are specific to each of the categories. When it comes to object detection, the problem of class-imbalance can pose a major hurdle in terms of the performance of object detection models. Object detection also involves identification of localized information (that is specific to different parts of an image) from the full image space. Therefore, the task of object detection is a combination of identification of localized information and classification [281]. In 2016, Hwang and Kim proposed a self-transfer learning (STL) framework which optimizes both the aspects of medical object detection task. They tested the STL framework for the detection of nodules in chest radiographs and lesions in mammography [210]. Yildrim et al. [219] proposed a deep transfer learning approach for automated diagnosis of diabetes mellitus. Heart rate (HR) signals were converted into frequency spectrum images after acquisition through Electrocardiogram (ECG) data. Subsequently, 2D-CNN models such as AlexNet, VggNet, ResNet, and DenseNet were applied on the frequency spectrum images wherein it was observed that the DenseNet pre-trained model had the highest average classification accuracy of 97.62% and a sensitivity of 100% in detecting diabetic subjects. In 2018, Raghavendra et al. [220] proposed an eighteen layer CNN framework for glaucoma diagnosis using 1426 fundus images (normal: 589 and glaucoma: 837) and obtained an accuracy of 98.13%, sensitivity of 98% and specificity of 98.3% across the board. In a later study in 2019, Raghavendra et al. [282] presented a computer-aided design (CAD) tool for highly precise detection of the presence of glaucoma by training a two-layer sparse autoencoder to learn meaningful features from fundus images. The proposed method resulted in an F-score of 0.95 across 1426 digital fundus images (837 glaucoma and 589 control) and is suggestive of the possibility of using the tool as a reference for verifying clinical inferences. Fujita et al. [283] also developed an automated computer-aided diagnosis (CAD) to identify impaired heart conditions such as fibrillations and flutters by analyzing the abnormal patterns in the signals from electrocardiogram (ECG) using deep convolutional neural networks. Acharya et al. [284] classified ECG beats between normal condition and with myocardial infarction using convolutional neural network and achieved an accuracy of 93.53% and 95.22% with and without noise in the signal respectively. In 2019, Acharya et al. [285] proposed a 11-layered deep convolutional neural network to diagnose congestive heart failures using ECG signals. Fujita et al. [286] proposed a 6-layer deep convolutional neural network to classify ECG patterns among normal and irregular conditions including atrial fibrillation (Afib), atrial flutter (Afl), and ventricular fibrillation. The model necessitates simple feature extraction without further pre-processing. Most recently, Gangwar and Ravi [218] proposed a hybrid model comprising transfer learning on pre-trained Inception-Resnet-v2 followed by a block of CNN layers. They reported a test accuracy of 72.33% and 82.18% on Messidor-1 and APTOS datasets respectively.

Segmentation is a widely researched domain within the purview of deep learning powered medical image processing. Organ and substructure segmentation allows for advanced fine-grained analysis of a medical image and it is widely practiced in the analyses of cardiac and brain images. Segmentation includes both the local and global context of pixels with respect to a given image. A demonstration is shown in Fig. 13, where the segmented parts of the brain tumor indicate various health conditions. The performance of a segmentation model can suffer from inconsistencies due to class imbalances, which makes the task a difficult one. A widely-used CNN architecture for medical image segmentation is the U-Net proposed by Ronneberger et al. [287] in 2015. U-Net takes care of sampling that is required to check the class-imbalance factors and it is capable of scanning an entire image in just one forward pass which enables it to consider the full context of the image. RNN-based architectures have also been proposed for segmentation tasks. In 2016, Andermatt et al. [211] presented a method to automatically segment 3D volumes of biomedical images. They used multi-dimensional gated recurrent units (GRU) as the main layers of their neural network model. The proposed method also involves on-the-fly data augmentation

which enables the model to be trained with less amount of training data.

Other applications of deep learning in medical image processing include image registration which implies coordinate transformation from a reference image space to target image space. Cheng et al. [212] used multi-modal stacked denoising autoencoders to compute effective similarity measure among images using normalized mutual information and local cross correlation. On the other hand, Miao et al. [213] developed CNN regressors to directly evaluate the registration transformation parameters. In addition to these, image generation and enhancement techniques have been discussed in [214] and [215].

The marriage of deep learning and medical image processing has thus far produced outstanding results across the domain. However in a regulation-sensitive domain such as this, Maier et al. [288] contend that prior knowledge should be incorporated in cases of image detection, recognition and reconstruction so that data-driven approaches do not produce implausible results.

5.5. Power systems

Artificial Neural Networks (ANN) have rapidly gained popularity among power system researchers [224]. Since their introduction to the power systems area in 1988 [225], numerous applications of ANN to problems of electric power systems have been proposed. However, the recent developments of Deep Learning (DL) methods have resulted into powerful tools that can handle large data-sets and often outperform traditional machine learning methods in problems related to the power sector [226]. For this reason, currently deep architectures are receiving the attention of researchers in power industry applications. Here, we will focus on describing some approaches of deep ANN architectures applied on three basic problems of the power industry, i.e. load forecasting and prediction of the power output of wind and solar energy systems.

Load forecasting is one of the most important tasks for the efficient power system's operation. It allows the system operator to schedule spinning reserve allocation, decide for possible interchanges with other utilities and assess system's security [227]. A small decrease in load forecasting error may result in significant reduction of the total operation cost of the power system [228]. Among the Artificial Intelligence techniques applied for load forecasting, methods based on ANN have undoubtedly received the largest share of attention [229]. A basic reason for their popularity lies on the fact that ANN techniques are well-suited for energy forecast [230]; they may obtain adequate estimations in cases where data is incomplete [231] and can consistently deal with complex non-linear problems [232]. Park et al. [233], was one of the first approaches for applying ANN in load forecasting. The efficiency of the proposed Multi-layer Perceptron (MLP) was demonstrated by benchmarking it against a numerical forecasting method frequently used by utilities. As an evolution of ANN forecasting techniques, DL methods are expected to increase the prediction accuracy by allowing higher levels of abstraction [234]. Thus, DL methods are gradually gaining increased popularity due to their ability to capture data behavior when considering complex non-linear patterns and large amounts of data. In [235], an end-to-end model based on deep residual neural networks is proposed for hourly load forecasting of a single day. The proposed DL architecture comprises a number of residual blocks, as proposed in [237], in which Scaled Exponential Linear Units (SELU) are used as activation function in all hidden layers. Two short-cut connections are also added; the former is used to bypass several adjacent residual blocks, while the latter connects the input to the outputs. Moreover, inspired by the structure of the CNN, the proposed network structure is further modified. In particular, a series of side residual blocks are included, whose input is the output of the first residual block of the main path. Only raw data of past load and temperature were used as inputs of the model. Initially, the inputs of the model are processed by several fully connected layers to produce preliminary forecast. These forecasts are then passed through the deep neural network structure. The efficiency of the proposed model was demonstrated on data-sets from the North-American Utility and ISO-NE. The results reveal that, the additional side residual blocks and the dense shortcut connections have improved the performance of the network compared to the basic architecture. Moreover, the proposed method has managed to outperform several existing models including the ensemble approach in [238] and the Radial Basis Function network of [239], in the aforementioned data sets. In particular, the method has provided load forecasting predictions with improvements of up to 8.9% with respect to the MAPE metric compared to the benchmark approaches. In [236], a Long Short Term Memory (LSTM)-based neural network has been proposed for short and medium term load forecasting. In order to optimize the effectiveness of the proposed approach, Genetic Algorithm is used to find the optimal values for the time lags and the number of layers of the LSTM model. The LSTM architecture that provided the best prediction accuracy comprised 6 hidden layers, while the activation function was the ReLU. The efficient performance of the proposed structure was verified using electricity consumption data of the France Metropolitan. The method was benchmarked against the Extra Tree Regressor method, which has provided the best performance amongst several machine learning techniques applied on the same data set. The results of the computational experiments demonstrate that the proposed DL approach has outperformed the benchmark algorithm on the examined test case with respect to RMSE, MAE and CV. Mocanu et al. [234] utilized two deep learning approaches based on Restricted Boltzman Machines (RBM), i.e. conditional RBM (CRBM) and factored conditional RBM (FCRBM), for single-meter residential load forecasting. In the former model the RBM are extended by including a conditional history layer. The FCRBM consists of the layers of the CRBM, i.e. the visible, hidden and history layers, as well as two additional layers for styles and features. Nevertheless, in the proposed method the styles and feature layers are combined to a single layer. The relations amongst these layers are modeled using undirected or directed weights and factors. The employed architectures were benchmarked against a shallow ANN, an RNN, and a SVM on data sets of energy consumption of a household. Both methods have shown increased efficiency compared to the competing algorithms for several scenarios of time resolution of prediction. Moreover, while the forecasting horizon increases the proposed methods are more robust and their prediction errors, in terms of RSME, is significantly lower compared to the corresponding values of a shallow ANN formulation. For example, when the aggregate energy consumption for a week is predicted with 15 min. interval the FCRBF exhibits a RMSE of 0.7971 while the corresponding of the shallow ANN is 1.8679. Dedinec et al. [240] employed a Deep Belief Network (DBN) for short term load forecasting of the Former Yugoslavian Republic of Macedonia. The proposed network comprised several stacks of RBM, which were pre-trained layer-wise. The unsupervised training of the RBMs is followed by a fine-tuning of the parameters of the model using a back-propagation algorithm for supervised training. The results of the method are compared to the latest actual data, to a typical feed forward multi-layer NN and to the results obtained from a traditional neural network model used by the system operator of FYROM for providing the 24 h ahead load forecast. The results reveal that the proposed method manages to decrease the MAPE by 8.6% compared to the traditional neural network model used

by the system operator for 24 h ahead load forecasting, and by 21% when the peak load of the following day is forecasted. Moreover in all the data sets examined the DBN has managed to provide lower MAPE compared to the shallow NN architecture for several configurations of neurons in the hidden layers. Rahman et al. [241] proposed two models based on the architecture of Recurrent Neural Networks (RNN) aiming to predict the medium and long term electricity consumption in residential and commercial buildings with one-hour resolution. The approach has utilized a MLP in combination with a LSTM based model using an encoder–decoder architecture. A model based on LSTM-RNN framework with appliance consumption sequences for short term residential load forecasting has been proposed in [242]. The researchers have showed that their method outperforms other state-of-the-art methods for load forecasting such as the conventional BPNN, the k-nearest Neighborhood regression, the Extreme Learning Machine, and the method based on a sophisticated input selection scheme combined with a hybrid forecasting framework. In particular, considering energy data from 69 households, the proposed method has managed to derive predictions with the lower MAPE value for more than 50% of them. For the case, where the aggregated load of the examined households is forecasted, the proposed method has provided the lowest forecasting error, with MAPE values ranging from 8.58% to 9.14%. Moreover, the results reveal that the higher the inconsistency is in daily consumption profiles of the households the more the forecasting accuracy of LSTM is increased compared to the conventional BPNN. In [243] a Convolutional Neural Network (CNN) with k-means clustering has been proposed. K-means is used to partition the large amount of data into clusters, which are then used to train the networks. The method has shown improved performance compared to the case where the k-means has not been engaged.

The utilization of DL techniques for modeling and forecasting in systems of renewable energy is progressively increasing. Since the data in such systems are inherently noisy, they may be adequately handled with ANN architectures [244]. Moreover, because renewable energy data is complicated in nature, shallow learning models may be insufficient to identify and learn the corresponding deep non-linear and non-stationary features and traits [245]. Among the various renewable energy sources, wind and solar energy have gained more popularity due to their potential and high availability [246]. As a result, in recent years the research endeavors have been focused on developing DL techniques for the problems related to the deployment of the aforementioned renewable energy sources.

Photovolatic (PV) energy has received much attention, due to its many advantages; it is abundant, inexhaustible and clean [247]. However, due to the chaotic and erratic nature of the weather systems, the power output of PV energy systems is intermittent, volatile and random [248]. These uncertainties may potentially degrade the real-time control performance, reduce system economics, and thus pose a great challenge for the management and operation of electric power and energy systems [249]. For these reasons, the accuracy of forecasting of PV power output plays a major role in ensuring optimum planning and modeling of PV plants. In [245] a deep neural network architecture is proposed for deterministic and probabilistic PV power forecasting. The deep architecture for deterministic forecasting comprises a Wavelet Transform and a deep CNN. To handle the PV power output data, which are commonly 1D data in the time domain, a deep CNN architecture is utilized. It consists of a 1D-data-to -2d-Image layer, an alternating convolutional layer and a pooling layer, a 2D-Image-to -1D-data layer and a logistic regression layer. The 1D-data-to -2d-Image layer and the 2D-Image-to -1D-data layer are responsible for data dimension transformation. Moreover, WT is used to decompose the original PV power data series into several sub-signals with less uncertainties, better behavior and outliners, facilitating their predictability. Moreover, the probabilistic PV power forecasting model combines the deterministic model and a spine Quantile Regression (QR) technique. The method has been evaluated on historical PV power datasets obtained from two PV farms in Belgium. The method has demonstrated high forecasting robustness and has outperformed a conventional BPNN and a SVM exhibiting lower values of MAPE, MAE, RMSE on all the examined cases. For example, on monthly 45-minutes-ahead PV power forecasting results in Limburg the method has exhibited average MAPE of 0.0385, while the conventional BPNN and SVM 0.0933 and 0.0748, respectively. In Gensler et al. [250], several deep network architectures, i.e. MLP, LSTM networks, DBN and Autoencoders, have been examined with respect to their forecasting accuracy of the PV power output. The performance of the methods is validated on actual data from PV facilities in Germany. The architecture that has exhibited the best performance is the Auto-LSTM network, which combines the feature extraction ability of the Autoencoder with the forecasting ability of the LSTM. In [251] an LSTM-RNN is proposed for forecasting the output power of solar PV systems. In particular, the authors examine five different LSTM network architectures in order to obtain the one with the highest forecasting accuracy at the examined data-sets, which are retrieved from two cities of Egypt. The network, which provided the highest accuracy is the LSTM with memory between batches.

With the advantages of non-pollution, low costs and remarkable benefits of scale, wind power is considered as one of the most important sources of energy [252]. ANN have been widely employed for processing large amounts of data obtained from data acquisition systems of wind turbines [253]. In recent years, many approaches based on DL architectures have been proposed for the prediction of the power output of wind power systems. In [254], a deep neural network architecture is proposed for deterministic wind power forecasting, which combines CNN and LSTM networks. The results of the model are further analyzed and evaluated based on the wind power forecasting error in order to perform probabilistic forecasting. The method has been validated on data obtained from a wind farm in China; it has managed to perform better compared to other statistical methods, i.e. ARIMA and persistence method, as well as artificial intelligence based techniques in deterministic and probabilistic wind power forecasting. Wang et al. [255] proposed a probabilistic wind power forecasting method based on Wavelet Transform, CNN and ensemble technique. The modified CNN architecture proposed comprises the following: (i) a 1D-data-to −2D-Image layer to convert the 1D wind power time series data into 2D Image for feature extraction, (ii) several building blocks consisting of a convolution layer and a sub-sampling layer used to extract the hidden features of the data, (iii) a 2D-Image-to −1D-data layer to reconvert the 2D Image into 1D vector and (iv) a logistic regression layer for the prediction. Their method was compared with the persistence method, a shallow BPANN and a SVM approach, on data sets collected from wind farms in China. The results validate that their method outperforms the benchmark approaches with respect to the Average Coverage Error (ACE), Interval Sharpness (IS) and the Continuous Ranking Probability score (CPRS) metrics. For example regarding the ACE metric, which measures the extent to which the prediction quantiles match the actual values, the proposed method has obtained an average value of 0.16%, while the corresponding values obtained by the persistence method, the BPANN with QR and the SVM with QR are 6.81%, 3.51% and 1.72%, respectively, thus the resultant ACE of the proposed approach are closer to the corresponding nominal confidence level. The methods superior performance may be attributed to the modified deep CNN architecture and

the ensemble technique; the former is able to capture the non-linear and non-stationary features of the different wind power frequencies provided by the WT, while the latter is able to mitigate the wind power uncertainties with respect to data noise. In [256] a DBN model in conjunction with the k-means clustering algorithm is proposed for wind power forecasting. The proposed approach demonstrated significantly increased forecasting accuracy compared to a BPANN and a Morlet wavelet neural network on data-sets obtained from a wind farm in Spain. Finally, in [257] an approach is proposed for wind power forecasting, which combines deep Autoencoders, DBN and the concept of transfer learning. In particular, an architecture consisting of 9 stacked under complete[1] and sparse deep AE is used as the base regressor. Then a meta-regressor follows which comprises a DBN. Exploiting an ensemble of regressors may increase the overall forecasting robustness of the model [257]. Moreover, including DL architectures as base and meta regressors may increase the ability of the approach to learn non-linear features of the input data, triggering improved forecasting accuracy. The method is tested on data-sets containing power measurement and meteorological forecast related to components of wind, obtained from wind farms in Europe. Moreover, it is compared to commonly used baseline regression models, i.e. ARIMA and Support Vector Regressor (SVR) as well as the methods of Grassi et al. [289], Amjady's et al. [290] and Zameer et al. [291]. The proposed method manages to outperform both the baseline regression models and the approaches in [289,290] and [291]. For example the SVR (with linear kernel) which was the baseline method with the best performance, has exhibited for wind farms 1 and 3 MAE values of 0.0722 and 0.1019, while the presented method has obtained 0.0658 and 0.0825, respectively.

5.6. Recommender systems

In recent years, the on-going developments on computer, networks, sensors and data storage technologies, has triggered a significant increase in the volume of available data [258]. Nevertheless the aforementioned comes at a cost for individuals; in the presence of excessive amount of information they may not be able to process them and utilize them adequately during decision making procedures. Recommender Systems (RS) have emerged as a very efficient tool to deal with this 'overload' of information. Such systems exploit a range of information provided by users, i.e. their behavior and/or their preferences, to establish a model, which recommends items and products which may be of interest to the users. With respect to the information utilized to create recommendations the three most common recommendation approaches are the following [259]:

- **Content based (CB):** The system learns to recommend items that are similar to the ones that the user liked in the past. Thus information from both the users profile and the description of the product are exploited.
- **Collaborative filtering (CF):** In this approach the system recommends to the active user the items that other users with similar preferences have selected in the past. The similarity in the 'taste' of two users may be calculated based on the similarity in their rating history or preference behavior.
- **Hybrid recommendation methods:** These RS are based on the combination of the above mentioned techniques, in an attempt to exploit and combine the advantages of each technique.

[1] Under complete is an AE in which the neurons in the hidden layer are less than those in the input layer.

Recently, DL techniques have been applied in the field of RS, facilitating the overcoming of obstacles of conventional models and achieving high recommendation quality. They are able to effectively capture the non-linear and complex relationship between the user and the item, enabling the modeling of non-trivial abstractions as data representations in the higher layers. In particular, researchers have attempted to utilize the efficacy of DL methods in extracting hidden features and relationships and have proposed several solutions to challenges faced by RS, i.e. accuracy, sparsity, cold start and scalability. The number of research publications on DL-based RS has significantly increased in recent years, evidently demonstrating the inevitable pervasiveness of deep learning in RS research. Several applications of deep learning models in the RS framework are described in the following, classified based on the DL model used in the RS.

Several RS have been proposed utilizing the Autoencoder in the relevant literature. AEs can be applied to build an RS either by learning a low dimensional representation of features or by directly providing the missing entries of rating matrix in the construction layer [260]. One of the first studies, utilizing AEs in RS is that of Sedhain et al. [261], where an AE-based collaborative filtering RS, referred as AutoRec, has been proposed. In this approach the original partial observed vectors have been replaced by integer ratings. The inputs of the model are the user-based ratings or the item-based ratings in the rating matrix R. Moreover, the model produces an output through encoding and decoding processes while the parameters of the model are optimized by minimizing the reconstruction error. Experimental results imply that AutoRec outperforms biased Matrix Factorization, RBM-based CF [56], and Local Low-rank Matrix Factorization (LLORMA) with respect to accuracy on MovieLens and Netflix data sets. Wu et al. [262] have included information related to item description in the approach of [261], in order to tackle problems related to cold-start. Wang et al. [263] have proposed a method which combines Stacked Denoising AEs and Probabilistic Matrix Factorization for improving the performance of rating prediction. In particular, a tightly coupled method for RS is proposed, which attempts to exploit a two-way interaction between the rating information and auxiliary information, such as item content information. This method, termed Collaborative DL (CDL), engages SDAEs and Collaborative Filtering method for the content information and the rating matrix, respectively. Computational experiments are carried out on two data sets from CiteULike and one data set from Netflix and recall is used as the performance metric. The method has managed to outperform several approaches including Collaborative Topic Regression, in which top modeling is performed for exploiting auxiliary information. For example, in the Netflix data series, the 2-layer CDL demonstrated improved performance compared to CTR by a margin of up to 5.9% in the sparse setting and 2.0% in the dense setting, signaling that the integration of DL in RS may lead to a significant performance improvement.

RBMs have been frequently exploited in the context of RS. In Salakhutdinov et al. [56], a model which combines RBM with Singular Value Decomposition (SVD) has been developed. In particular, undirected two layer RBMs are used to model table data and the ratings are represented by utilizing a one-hot-vector. It has been demonstrated that the proposed approach may provide more accurate predictions (by approximately 6%) compared to the conventional Netflix recommendation system. Georgiev et al. [264] have proposed an RS approach based on CF which utilizes RBM. In particular, the authors model both the user–user correlation and the item–item correlation in a hybrid framework. Moreover, real values are employed in the visible layer in contrast to multinomial variables, utilizing thus the natural order amongst user–item ratings. The proposed methods has been evaluated

on two MovieLens data sets of different sizes, i.e. MovieLens 100k and MovieLens 1M, significantly outperforming previously proposed CF approaches with respect to the MAE metric. Liu et al. [265] have proposed an improved Item Category aware Conditional Restricted Boltzmann Machine Frame model for recommendation, in an attempt to deal with the problem of large and highly sparse data sets. The visible layer of the RBM consists the rating matrix, while the hidden layer comprises a hidden feature vector. Item category information has been included as the conditional layer of the Conditional RBM, to increase the accuracy of the model. Two distinct models with respect to the difference in the visible layer representation are utilized. The former, i.e. IC-CRBMF item-based model, employs item's rating vectors and the conditional layer is the item category. The latter, i.e. IC-CRBMF user-based, exploits the given item's genre feature vector in the conditional layer. Both models have been tested on the MovieLens 100k and MovieLens 1M, and have demonstrated improved performance with respect to RMSE and MAE metrics compared to basic user-Based and item-Based CF and several methods based on AE for RS. For example in the MovieLens 1M data set, the IC-CRBMF item-based model has obtained $RMSE = 0.867$ and $MAE = 0.681$ values, which are significantly lower compared to the $RMSE = 0.902$ and $MAE = 0.699$ of the basic item-based CF.

RS based on DBN have been also developed. DBNs can be utilized in the context of RS to reduce the dimensionality or extract high level features from sparse data, tackling thus the cold-start problem. Hongliang and Xiaona [266] have proposed an RS for video recommendation, which deploys the common DBN and a CF algorithm, to alleviate the cold start problem. In particular, features of the user profile and the neighborhood relationships are extracted using the DBN, while the user-based CF is used to fill the missing ratings in the rating matrix. The method is applied on the Netflix data sets and has outperformed with respect to the MAE two RS based on the conventional Content-based and Collaborative Filtering, respectively. X. Wang and Y. Wang [267] have developed a content-based RS for music recommendation based on DBN and probabilistic graphical modeling. More specifically, the authors propose a unify approach, in which DBN is used for extracting features from audio content and making personalized recommendations. Subsequently, a hybrid model is proposed in which CF is combined with audio content by utilizing automatically learned audio features. The methods are tested on the Echo Nest Taste Profile Subset and the performance of the proposed method on the warm start stage and the cold-start stage is compared with the corresponding performance of two content-based approaches presented in [268], i.e. CB1 and CB2. The proposed method outperforms CB1 and CB2 in both stages with respect to the RMSE metric. For example, the RMSE of the proposed methods on the validation and test sets of the cold start stage were 0.477 and 0.478, respectively, while the corresponding values obtained by CB2 were 0.495 and 0.495. Several research endeavors have been presented in which the RS integrates CNN. In most such approaches CNN is commonly utilized for feature extraction. Zheng et al. [269] developed a deep model to learn item properties and user behaviors from reviews written by users. In particular, they combined two parallel CNN; the former learns the user's behavior taking advantage of the reviews written by the user, while the latter learns the properties of the item based on the relevant reviews. The model employs a word embedding method for mapping the textual content into a lower dimensional semantic space. The output of the two parallel networks is used as input for the prediction layer where the factorization machine is employed to predict the ratings of users for the examined items. Kim et al. [270] developed ConvMF, which is a context-aware RS; it combines a CNN model and Probabilistic Matrix Factorization in an attempt to effectively utilize both collaborative information and contextual information. The main objective is to address the issue of data sparsity for improving the performance of RS. Specifically, the algorithm utilizes CNN to capture contextual information of item description documents using word embeddings and convolutional filters. The method has been tested on three datasets, i.e. 1M MovieLens, 10M MovieLens and Amazon Instant Video. In this datasets the ConvMF has managed to improve the results with respect to the RMSE metric by 3.92%, 2.79% and 16.60%, respectively, compared to CDL method of [263], which combines PMF with SDAE.

Finally, some research efforts have focused on integrating MLPs within RS. He et al. [271] have proposed a framework named Neural network Collaborative Filtering (NCF). In this model, the MLP is utilized to replace MF and learn the user–item interaction function. The method focuses on implicit feedback, which reflects user preferences indirectly through examining behaviors like watching videos or clicking items. The authors demonstrate that MF can be an instant of NCF which can be significantly improved, when an MLP is utilized, due to the ability of approximating high non-linear features of the user–item relationship. In the MLP, the bottom input layer comprises two feature vectors of the user and the item, respectively; in their work the identity of a user and an item are the input feature thus the vectors are transformed into binary sparse vectors. Their method has been applied on data sets of MovieLens and Pinterest and has been benchmarked against state of the art RS for CF which utilize MF. The NCF framework combining MLP and Generalized MF manages to significantly outperform such methods with relative improvements close to 5% on both data sets. Tay et al. [272] have proposed a new neural architecture for recommendation with reviews. In particular, their model utilizes a MLP for multi-pointer learning schemes, which extracts important reviews from user and item reviews and combines then in a word-by-word fashion. Moreover, the authors propose a multi-pointer learning scheme which learns to combine multiple views of user–item interactions. The pointer setting of the model extracts the named reviews for direct review to review matching for learning of the item/user representations. The efficacy of the proposed method is examined on data sets of Amazon and Yelp. The results of the computational experiments demonstrate that the proposed method performs significantly better than state-of-the-art methods, achieving relative improvements of up to 71% compared to previously proposed method, such as DeepCoNN [269].

6. Discussions

In this paper several deep learning architectures, from the very foundational ones to the more recent developments have been briefly reviewed keeping in mind their continuous evolution over time as well as their applications to specific domains. A list of category-wise publicly available data repositories is aggregated in Table 4 for enthusiasts. A concise discussion on the blend of swarm intelligence in deep learning approaches is put forward and how the influence of one enriches the other when applied to real world problems is detailed. The breakneck speed at which the use of deep learning technologies is growing, specially in safety critical systems, brings us to the question of how robust the models are in the presence of adversarial scenarios. On this note, an overview of testing neural network architectures, the procedures for adversarial test generation as well as the countermeasures to be adopted against adversarial examples are reviewed. Advances in recent years across six application areas including Prognostics and Health Management, Financial Services, Financial Time Series Forecasting, Medical Image Processing, Power Systems and Recommender Systems are explored.

Table 4
A collection of data repositories for deep learning practitioners.

Category	Dataset	Link
Image Datasets	MNIST	http://yann.lecun.com/exdb/mnist/
	CIFAR-100	http://www.cs.utoronto.ca/~kriz/cifar.html
	Caltech 101	http://www.vision.caltech.edu/Image_Datasets/Caltech101/
	Caltech 256	http://www.vision.caltech.edu/Image_Datasets/Caltech256/
	Imagenet	http://www.image-net.org/
	COIL100	http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php
	STL-10	http://www.stanford.edu/~acoates//stl10/
	Google Open images	https://ai.googleblog.com/2016/09/introducing-open-images-dataset.html
	Labelme	http://labelme.csail.mit.edu/Release3.0/browserTools/php/dataset.php
Speech Datasets	Google Audioset	https://research.google.com/audioset/dataset/index.html
	TIMIT	http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1
	VoxForge	http://www.voxforge.org/
	CHIME	http://spandh.dcs.shef.ac.uk/chime_challenge/data.html
	2000 HUB5 English	https://catalog.ldc.upenn.edu/LDC2002T43
	LibriSpeech	http://www.openslr.org/12/
	VoxCeleb	http://www.robots.ox.ac.uk/~vgg/data/voxceleb/
	Open SLR	https://www.openslr.org/51
	CALLHOME American English Speech	https://catalog.ldc.upenn.edu/LDC97S42
Text Datasets	English Broadcast News	https://catalog.ldc.upenn.edu/LDC97S44
	SQuAD	https://rajpurkar.github.io/SQuAD-explorer/
	Billion Word Dataset	http://www.statmt.org/lm-benchmark/
	20 Newsgroups	http://qwone.com/~jason/20Newsgroups/
	Google Books Ngrams	https://aws.amazon.com/datasets/google-books-ngrams/
	UCI Spambase	https://archive.ics.uci.edu/ml/datasets/Spambase
	Common Crawl	http://commoncrawl.org/the-data/
	Yelp Open Dataset	https://www.yelp.com/dataset
Natural Language Datasets	Web 1T 5-gram	https://catalog.ldc.upenn.edu/LDC2006T13
	Blizzard Challenge 2018	https://www.synsig.org/index.php/Blizzard_Challenge_2018
	Flickr personal taxonomies	https://www.isi.edu/~lerman/downloads/flickr/flickr_taxonomies.html
	Multi-Domain Sentiment Dataset	http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
	Enron Email Dataset	https://www.cs.cmu.edu/~./enron/
	Blogger Corpus	http://u.cs.biu.ac.il/~koppel/BlogCorpus.htm
	Wikipedia Links Data	https://code.google.com/archive/p/wiki-links/downloads
	Gutenberg eBooks List	http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs
	SMS Spam Collection	http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/
	UCI's Spambase data	https://archive.ics.uci.edu/ml/datasets/Spambase
Geospatial Datasets	OpenStreetMap	https://www.openstreetmap.org
	Landsat8	https://landsat.gsfc.nasa.gov/landsat-8/
	NEXRAD	https://www.ncdc.noaa.gov/data-access/radar-data/nexrad
	ESRI Open data	https://hub.arcgis.com/pages/open-data
	USGS EarthExplorer	https://earthexplorer.usgs.gov/
	OpenTopography	https://opentopography.org/
	NASA SEDAC	https://sedac.ciesin.columbia.edu/
	NASA Earth Observations	https://neo.sci.gsfc.nasa.gov/
	Terra Populus	https://terra.ipums.org/
Recommender Systems Datasets	Movielens	https://grouplens.org/datasets/movielens/
	Million Song Dataset	https://www.kaggle.com/c/msdchallenge
	Last.fm	https://grouplens.org/datasets/hetrec-2011/
	Book-crossing Dataset	http://www2.informatik.uni-freiburg.de/~cziegler/BX/
	Jester	https://goldberg.berkeley.edu/jester-data/
	Netflix Prize	https://www.netflixprize.com/
	Pinterest Fashion Compatibility	http://cseweb.ucsd.edu/~jmcauley/datasets.html#pinterest
	Amazon Question and Answer Data	http://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_qa
	Social Circles Data	http://cseweb.ucsd.edu/~jmcauley/datasets.html#socialcircles
Economics and Finance Datasets	Quandl	https://www.quandl.com/
	World Bank Open Data	https://data.worldbank.org/
	IMF Data	https://www.imf.org/en/Data
	Financial Times Market Data	https://markets.ft.com/data/
	Google Trends	https://trends.google.com/trends/?q=google&ctab=0&geo=all&date=all&sort=0
	American Economic Association	https://www.aeaweb.org/resources/data/us-macro-regional
	US stock Data	https://github.com/eliangcs/pystock-data
	World Factbook	https://www.cia.gov/library/publications/download/
	Dow Jones Index Data Set	http://archive.ics.uci.edu/ml/datasets/Dow+Jones+Index
Autonomous Vehicles Datasets	BDD100k	https://bdd-data.berkeley.edu/
	Baidu Apolloscapes	http://apolloscape.auto/
	Comma.ai	https://archive.org/details/comma-dataset
	Oxford's Robotic Car	https://robotcar-dataset.robots.ox.ac.uk/
	Cityscape Dataset	https://www.cityscapes-dataset.com/
	CSSAD Dataset	http://aplicaciones.cimat.mx/Personal/jbhayet/ccsad-dataset
	KUL Belgium Traffic Sign Dataset	http://www.vision.ee.ethz.ch/~timofter/traffic_signs/
	LISA	http://cvrr.ucsd.edu/LISA/datasets.html
	Bosch Small Traffic Light	https://hci.iwr.uni-heidelberg.de/node/6132
	LaRa Traffic Light Recognition	http://www.lara.prd.fr/benchmarks/trafficlightsrecognition
	WPI Datasets	http://computing.wpi.edu/dataset.html

For each application area current research trends are dissected as well as feasible research directions outlined. Table 5 lists a collection of recent reviews across different application areas of deep learning including Computer Vision, Forecasting, Image Processing, Adversarial Cases, Autonomous Vehicles, Natural Language Processing, Recommender Systems and Big Data Analytics.

- **Comparison with some existing reviews**
  In keeping with the aim to capture information not present in detail in extant reviews on the subject, this work presents separate sections that fill in gaps and extend the chain of information. For example, Guo et al.'s review [292] looks at image classification, object detection, image retrieval and semantic segmentation as well as human pose estimation but does not introduce their applications in medical image analysis for disease identification. This review has a separate section on this very topic. Voulodimos et al. [293] dissected various applications of DNN to image processing tasks but pointed to only a few recent references in medical imaging, whereas this review again, has a dedicated section discussing recent strides in the domain. Kamilaris and Francesc [294] put forward a comprehensive review dedicated to deep learning in agriculture, whereas this work is more general in nature which provides a foundational introduction to deep learning and covers six different domains of application. A thorough exposition of the interplay between deep learning and big data by Zhang et al. [295] surveys DNN architectures applied to four dimensions of big data viz. voluminous data, heterogeneous data, real-time data and low quality data. In contrast this review is exploratory in nature and is not focused particularly on any one aspect of big data research. Rather, it is semantically different in that it summarizes the major findings using big data across a multitude of active research niches.
- **Subtle differences between this review and previous ones**
  The thrust has been to set the beginner up with a basic understanding of deep neural networks and how representation learning is accomplished at scale using such networks as well as to provide an advanced reader with a concise summary of some recent architectures before leading on to pertinent applications and highlighting some interesting results from influential papers in recent times. In terms of the major difference between this work and the related previous ones, following points may be noted:
  For the beginner: This survey handholds the interested reader from being an enthusiast to being ready to delve deep into the DL paradigm. Specifically the introductory Section 1.1, 1.2 and 1.3 have been purposefully inserted to create this non-existent bridge in the other reviews and Tables 1 and 2 provide sufficient starting points for someone interested in the subject. Without a proper understanding of why *'deep'* networks work and why they work now as opposed to before, the organization of an exploratory review would be ad-hoc.
  Coverage of advanced topics: We have invested in topics that are garnering interest now: Section 2.8 on Recent Deep Architectures, Section 3 on Swarm Intelligence in Deep Learning, Section 4 on Testing Deep Neural Networks are research directions not addressed adequately in other reviews. Particularly in Section 4, we introduce authors to Adversarial Machine Learning, a hot research topic over the last five years.
  Survey of six active areas of research: The applications of deep learning in Section 5 follows the research interest of the authors to impart subject matter expertise. Areas such as fraud detection in financial services, financial time-series forecasting, power systems, prognostics and health management, medical image processing and recommender systems are brought under the purview of one review and the focus has been on discussing key articles from the past few years and organizing them for handy reference in Table 5. In fact, useful reviews from 8 different application areas that are gaining momentum as of date have been listed and linked.
  Material for further reading and references: A rich list of data repositories is provided in Table 4 aimed to assist in following up a research problem in a particular niche. Specifically, around 75 data repositories spanning the following 8 areas are listed: (a) Image Datasets, (b) Speech Datasets, (c) Text Datasets, (d) Natural Language Datasets, (e) Geospatial Datasets, (f) Recommender Systems Datasets, (g) Economics and Finance Datasets, (h) Autonomous Vehicles Datasets.
  In synopsis, this review takes an interested reader through a journey of how deep learning has emerged as a game changing technology and provides them with adequate pointers to get started. It then makes advanced learning possible through a concise summary of important architectures and recent developments before finally highlighting the recent advancements in six key areas. The review is geared towards making an introduction to the field as well as for people who want to skip to sections that are of interest to them.

7. Conclusions and future research directions

This review presents an overview of the various opportunities and challenges in deep learning which serve as potential directions for future research. In conclusion, a few such open areas of research are highlighted and some of the existing lines of thoughts and studies in addressing related challenges are elaborated.

- **Challenges with scarcity of data:** With growing availability of data as well as powerful and distributed processing units, Deep Learning architectures can be successfully applied to major industrial problems. However, deep learning is traditionally big data driven and lacks efficiency to learn abstractions through clear verbal definitions [344] if not trained with large training sample sizes. Also the large reliance on Convolutional Neural Networks (CNNs) especially for video recognition purposes could face exponential inefficiency leading to their demise [345] which can be avoided by capsules [346] capturing critical spatial hierarchical relationships more efficiently than CNNs with lesser data requirements. To make DL work with smaller available data sets, some of the approaches in use are data augmentation, transfer learning, recursive classification techniques as well as synthetic data generation. One shot learning [347] is also bringing in new avenues to learn from very few training examples which has already started showing progress in language processing and image classification tasks. More generalized techniques are being developed in this domain to make DL models learn from sparse or fewer data representations.
- **Data preprocessing overheads:** Sensor fusion has rendered a data explosion in recent times and while more data equates to more training examples, an important issue for machine learning practitioners is to separate the good from the bad. While data may be sourced in structured or semi-structured ways, filtering out bad instances and/or instances which are uncorrelated to the learning objective still remains a key challenge requiring further research. The *quality* of the training data is dependent among other things, on ensuring missing/incorrect values and invalid/wrong format

Table 5
A collection of recent reviews on deep learning.

Topic	Review papers
Computer Vision	Deep learning for visual understanding: A review [292] Deep learning for computer vision: A brief review [293] A survey on deep learning methods for robot vision [296] Deep learning advances in computer vision with 3D data: A survey [297] Visualizations of deep neural networks in computer vision: A survey [298]
Forecasting	Machine learning in financial crisis prediction: A survey [299] Deep learning for time-series analysis [300] A survey on machine learning and statistical techniques in bankruptcy prediction [301] Time series forecasting using artificial neural networks methodologies: A systematic review [302] A review of deep learning methods applied on load forecasting [303] Trends in machine learning applied to demand & sales forecasting: A review [304] A survey on retail sales forecasting and prediction in fashion markets [305] Electric load forecasting: Literature survey and classification of methods [306] A review of unsupervised feature learning and deep learning for time-series modeling [307]
Image Processing	A survey on deep learning in medical image analysis [216] A comprehensive survey of deep learning for image captioning [308] Biological image analysis using deep learning-based methods: Literature review [309] Deep learning for remote sensing image classification: A survey [310] Deep convolutional neural networks for image classification: A comprehensive review [311] Deep learning for medical image processing: overview, challenges and the future [312] An overview of deep learning in medical imaging focusing on MRI [313] Deep learning in medical ultrasound analysis: A review [314]
Adversarial Cases	Threat of adversarial attacks on deep learning in computer vision: A survey [315] Adversarial learning in statistical classification: A comprehensive review of defenses against attacks [316] Adversarial attacks and defenses against deep neural networks: A survey [317] Adversarial machine learning: A literature review [318] Review of artificial intelligence adversarial attack and defense technologies [319] A survey of adversarial machine learning in cyber warfare [320]
Autonomous Vehicles	Planning and decision-making for autonomous vehicles [321] A review of deep learning methods and applications for unmanned aerial vehicles [322] MIT autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation [323] Perception, planning, control, and coordination for autonomous vehicles [324] Survey of neural networks in autonomous driving [325] Self-driving cars: A survey [326]
Natural Language Processing	Recent trends in deep learning based natural language processing [327] A survey of the usages of deep learning in natural language processing [328] A survey on the state-of-the-art machine learning models in the context of NLP [329] Inflectional review of deep learning on natural language processing [330] Deep learning for natural language processing: advantages and challenges [331] Deep learning for natural language processing [332]
Recommender Systems	Deep learning based recommender system: A survey and new perspectives [333] A survey of recommender systems based on deep learning [334] A review on deep learning for recommender systems: challenges and remedies [335] Deep learning methods on recommender system: A survey of state-of-the-art [336] Deep learning-based recommendation: Current issues and challenges [337] A survey and critique of deep learning on recommender systems [338]
Big Data Analytics	Efficient machine learning for big data: A review [339] A survey on deep learning for big data [295] A survey on data collection for machine learning: a big data – AI integration perspective [340] A survey of machine learning for big data processing [341] Deep learning in big data Analytics: A comparative study [342] Deep learning applications and challenges in big data analytics [343]

representations are at a minimum. *Sparsity* as an issue has a significant impact on the final accuracy of the model. Denser datasets usually lead to better performance, since there are more data points for the model to train from [348]. Missing values in data may be entirely random and independent of unobserved and observed variables (MCAR — Missing Completely At Random) or not random (MNAR — Missing Not At Random) where there is a dependency between missing feature values and the reason it is missing [349]. In the MAR (Missing At Random) assumption, the missing feature values may not be random and may be completely accounted for given sufficient information. Missing data can lead to sub par representation learning and render incorrect interpretations about the underlying patterns but there are data analysis methods that are robust to missing values, meaning that mild departure from the assumptions typically have little to no effect (bias, distortion, etc.) on the inference of the model. Methods such as imputation, interpolation (linear, bi-linear etc.), data deletion (listwise, pairwise, etc.), generative (Expectation Maximization, Full Information Maximum Likelihood estimation (FIML), etc.) and discriminative (maximum margin classification with absent features [350,351]) are well-known in dealing with such cases. The newer approaches such as Full Information Maximum Likelihood (FIML), Multiple Imputation (MI) and Expectation Maximization (EM) account for the conditions in which missing data is logged and are able to outperform parameter estimation when compared to listwise and pairwise data deletion methods [352]. In addition, model based techniques [353,354] offer a robust way of checking if the missing data is MCAR, MAR or MNAR. Pre-processing the data before feeding it to a network adds significant

overheads and necessitates further research as challenges involved keep scaling with the volume of data acquisition.

- **Adopting metamodeling approaches:** The combination of deep learning with unsupervised learning methods has become a popular research direction. Systems developed to set their own goals [344] and develop problem-solving approaches while exploring an environment are yielding interesting findings and hint at good efficacy while somewhat eliminating the need for data-hungry supervised learning. In particular, Meta Learning as a philosophy i.e. learning to learn is gaining momentum. The approach involves automated model design and decision making capabilities and optimizes the ability to learn a multitude of tasks from fewer training data [355]. Typically, meta-learning systems learn by being subjected to a large number of jobs and are tested on their ability to learn these new jobs. During meta-learning the model is trained to learn jobs using a meta-training set and has two optimizations at its core: (a) the *learner* which learns a job and (b) the *meta-learner* or *trainer*, which trains the learner [356].
- **Influence on cognitive neuroscience:** Inspiration drawn from cognitive neuroscience and developmental psychology in deciphering human behavioral patterns are able to bring major breakthroughs in applications such as enabling artificial agents to learn about spatial navigation on their own which comes naturally to most living beings, as demonstrated by Banino et al. [357]. This supports theories from neuroscience that establish grid cells as a critical component for vector-based navigation. The work contends that the latter can be coupled with path-based strategies to aid in navigation in challenging environments [358–360]. Lacking scientific explanations of how human beings solve complex visual and auditory problems, a section of cognitive scientists have embraced deep neural networks as models of human brain responses and behavior [361]. Given the ongoing debate in the cognitive science community regarding the use of deep learning, Cichy and Kaiser [361] opine on among other things, the particular question - "...are DNNs only valuable as predictive tools or do they also offer useful explanations of the phenomena investigated?"
- **Neural networks and reinforcement learning:** Metamodeling approaches using Reinforcement Learning (RL) are being used for designing problem specific Neural Network architectures. In [362] the authors introduced MetaQNN, a RL based meta-modeling algorithm to automatically generate CNN architectures for image classification by using Q-learning [363] with ϵ greedy exploration. AlphaGo [364], the computer program built combining reinforcement learning and CNN for playing the game 'Go' achieved a great success by beating human professional 'Go' players. Also deep convolutional neural networks can work as function approximators to predict 'Q' values in a reinforcement learning problem. So, a major thrust of current research is on superposition of neural networks and reinforcement learning geared towards problem specific requirements.
- **Creating general multi-purpose architectures:** Despite the success of Deep learning techniques across several domains, most of the neural architectures used so far are specifically trained to learn and perform a particular job and not a variety of disparate jobs at once. One of the major research challenges is to create general multi-purpose architectures that can encapsulate the learning from different domains as well as have the flexibility of transfer learning. To this end, Kaiser et al. [365] proposed a single model that has been trained simultaneously for a variety of tasks including image classification and captioning, translation, speech recognition and language parsing tasks. The model contains convolutional as well as sparsely gated layers along with attention mechanism acting together as subsets of the overall neural architecture and possesses demonstrated capability of learning multiple tasks jointly from different domains. This kind of multi-task learning capabilities when extended to various other real-world problems can bring significant performance improvement [366].
- **Using Incremental Learning techniques:** One of the most difficult challenges in financial markets is to trade in real time. An algorithm must be able to track any change and make good trades within a specific time frame [367]. However, when the dynamics of the dataset change, any previously trained model might perform poorly. In order for an algorithm to keep performing well, it should be trained repetitively. Deep learning algorithms/architectures, however, take longer and more resources to train. This might lead to investors missing out on many good trades which will lead to profits. The application of DL with faster incremental learning [368] in the financial market especially for automated trading will go a long way for investors to rely on DL models. Also, Progressive Neural Networks [369] are able to learn continually by accumulating knowledge and transferring them to a new domain. They are immune to catastrophic forgetting and can leverage previously gathered knowledge from pre-trained models via lateral connections and continually learn new tasks and thus has been shown to perform well in reinforcement learning tasks, such as playing Atari and 3D maze games [369].

This review is aimed at helping the beginner as well as the advanced practitioner make informed choices and has charted an in-depth analysis of the most recent deep learning architectures. It has gone a step further to render an exploratory dissection of some of the most active application areas in deep learning. The expectation is that readers will find the surveyed material engaging, yet easy to understand thereby staying abreast of the newest developments in this exciting field that holds such promise for the future.

CRediT authorship contribution statement

Saptarshi Sengupta: Conceptualization, Formal analysis, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Sanchita Basak:** Conceptualization, Investigation, Methodology, Project Administration, Resources, Software, Visualization, Writing - original draft, Writing - review & editing. **Pallabi Saikia:** Investigation, Resources, Software, Visualization, Writing - original draft, Writing - review & editing. **Sayak Paul:** Investigation, Resources, Software, Visualization, Writing - original draft. **Vasilios Tsalavoutis:** Investigation, Resources, Software, Visualization, Writing - original draft, Writing - review & editing. **Frederick Atiah:** Investigation, Resources, Software, Visualization, Writing - original draft, Writing - review & editing. **Vadlamani Ravi:** Conceptualization, Formal analysis, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing - original draft, Writing - review & editing. **Alan Peters:** Investigation, Resources, Software, Visualization, Writing - original draft.

References

[1] M. van Gerven, S. Bohte, Editorial: Artificial neural networks as models of neural information processing, Front. Comput. Neurosci. 11 (2017) 114, http://dx.doi.org/10.3389/fncom.2017.00114, URL https://www.frontiersin.org/article/10.3389/fncom.2017.00114.

[2] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133, http://dx.doi.org/10.1007/BF02478259.

[3] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436.

[4] S. Lawrence, C.L. Giles, A.D. Back, Face recognition: a convolutional neural-network approach, IEEE Trans. Neural Netw. 8 (1) (1997) 98–113, http://dx.doi.org/10.1109/72.554195.

[5] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440, http://dx.doi.org/10.1109/CVPR.2015.7298965.

[6] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 677–691, http://dx.doi.org/10.1109/TPAMI.2016.2599174.

[7] X. Wu, R. He, Z. Sun, A lightened CNN for deep face representation, 2015, CoRR abs/1511.02683. arXiv:1511.02683. URL http://arxiv.org/abs/1511.02683.

[8] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, L.V. Gool, Weakly supervised cascaded convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5131–5139, http://dx.doi.org/10.1109/CVPR.2017.545.

[9] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, K. Wang, J. Yan, C. Loy, X. Tang, Deepid-net: Object detection with deformable part based convolutional neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (7) (2017) 1320–1334, http://dx.doi.org/10.1109/TPAMI.2016.2587642.

[10] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Systems 2 (4) (1989) 303–314, http://dx.doi.org/10.1007/BF02551274.

[11] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (2) (1991) 251–257, http://dx.doi.org/10.1016/0893-6080(91)90009-T, URL http://www.sciencedirect.com/science/article/pii/089360809190009T.

[12] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The expressive power of neural networks: A view from the width, 2017, CoRR abs/1709.02540. arXiv:1709.02540. URL http://arxiv.org/abs/1709.02540.

[13] B. Hanin, Universal function approximation by deep neural nets with bounded width and relu activations, 2017, URL https://arxiv.org/abs/1708.02691.

[14] J. Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117, http://dx.doi.org/10.1016/j.neunet.2014.09.003, URL http://www.sciencedirect.com/science/article/pii/S0893608014002135.

[15] G. Marcus, Deep learning: A critical appraisal, 2018, arXiv e-prints arXiv:1801.00631.

[16] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, A. Swami, The limitations of deep learning in adversarial settings, in: 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016, pp. 372–387, http://dx.doi.org/10.1109/EuroSP.2016.36.

[17] E. Abbe, C. Sandon, Provable limitations of deep learning, 2018, arXiv e-prints arXiv:1812.06369.

[18] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychol. Rev. (1958) 65–386.

[19] and, Madaline rule ii: a training algorithm for neural networks, in: IEEE 1988 International Conference on Neural Networks, Vol. 1, 1988, pp. 401–408, http://dx.doi.org/10.1109/ICNN.1988.23872.

[20] B. Widrow, M.A. Lehr, 30 years of adaptive neural networks: perceptron, madaline, and backpropagation, Proc. IEEE 78 (9) (1990) 1415–1442, http://dx.doi.org/10.1109/5.58323.

[21] M. Minsky, S. Papert, Perceptrons - an introduction to computational geometry, 1969.

[22] P.J. Werbos, The roots of backpropagation: from ordered derivatives to neural networks and political forecasting, Vol. 1, John Wiley & Sons, 1994.

[23] J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. 79 (8) (1982) 2554–2558, http://dx.doi.org/10.1073/pnas.79.8.2554, arXiv:https://www.pnas.org/content/79/8/2554.full.pdf, URL https://www.pnas.org/content/79/8/2554.

[24] D.E. Rumelhart, G.E. Hinton, R.J. Williams, in: D.E. Rumelhart, J.L. McClelland, C. PDP Research Group (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 318–362, URL http://dl.acm.org/citation.cfm?id=104279.104293.

[25] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297, http://dx.doi.org/10.1023/A:1022627411411.

[26] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735, arXiv:https://doi.org/10.1162/neco.1997.9.8.1735, URL https://doi.org/10.1162/neco.1997.9.8.1735.

[27] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[28] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (7) (2006) 1527–1554.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous systems, 2015, Software available from tensorflow.org, URL http://tensorflow.org/.

[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, CaFfe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22Nd ACM International Conference on Multimedia, in: MM '14, ACM, New York, NY, USA, 2014, pp. 675–678, http://dx.doi.org/10.1145/2647868.2654889, URL http://doi.acm.org/10.1145/2647868.2654889.

[32] S. Tokui, K. Oono, S. Hido, J. Clayton, Chainer: a next-generation open source framework for deep learning, in: Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Twenty-Ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015, URL http://learningsys.org/papers/LearningSys_2015_paper_33.pdf.

[33] F. Chollet, et al., Keras, 2015, URL https://github.com/fchollet/keras.

[34] J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang, X. Jia, C. Zhang, Y. Wan, Z. Li, J. Wang, S. Huang, Z. Wu, Y. Wang, Y. Yang, B. She, D. Shi, Q. Lu, K. Huang, G. Song, Bigdl: A distributed deep learning framework for big data, 2018, CoRR abs/1804.05839.

[35] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, 2016, arXiv e-prints abs/1605.02688, URL http://arxiv.org/abs/1605.02688.

[36] F. Seide, A. Agarwal, CNTK: Microsoft's open-source deep-learning toolkit, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '16, ACM, New York, NY, USA, 2016, p. 2135, http://dx.doi.org/10.1145/2939672.2945397, URL http://doi.acm.org/10.1145/2939672.2945397.

[37] S. Kombrink, T. Mikolov, M. Karafiát, L. Burget, Recurrent neural network based language modeling in meeting recognition, in: Twelfth Annual Conference of the International Speech Communication Association, 2011.

[38] L. Deng, D. Yu, et al., Deep learning: methods and applications, Found. Trends Signal Process. 7 (3–4) (2014) 197–387.

[39] Y. Bengio, et al., Learning deep architectures for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[40] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[41] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in Neural Information Processing Systems, 2007, pp. 153–160.

[42] Y. Bengio, N. Boulanger-Lewandowski, R. Pascanu, Advances in optimizing recurrent networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 8624–8628.

[43] G.E. Dahl, T.N. Sainath, G.E. Hinton, Improving deep neural networks for lvcsr using rectified linear units and dropout, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 8609–8613.

[44] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015, arXiv preprint arXiv:1502.03167.

[45] D. Sussillo, L. Abbott, Random walk initialization for training very deep feedforward networks, 2014, arXiv preprint arXiv:1412.6558.

[46] D. Mishkin, J. Matas, All you need is a good init, 2015, arXiv preprint arXiv:1511.06422.

[47] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.

[48] S.K. Kumar, On weight initialization in deep neural networks, 2017, arXiv preprint arXiv:1704.08863.

[49] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. Icml, 2013, Vol. 30,

[50] A. Fischer, C. Igel, An introduction to restricted Boltzmann machines, in: Iberoamerican Congress on Pattern Recognition, Springer, 2012, pp. 14–36.

[51] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, Tech. rep., COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

[52] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (8) (2002) 1771–1800.

[53] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.

[54] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[55] H. Larochelle, Y. Bengio, Classification using discriminative restricted boltzmann machines, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 536–543.

[56] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: Proceedings of the 24th International Conference on Machine Learning, ACM, 2007, pp. 791–798.

[57] J. Bennett, S. Lanning, et al., The netflix prize, 2007.

[58] I. Sutskever, G. Hinton, Learning multilevel distributed representations for high-dimensional sequences, in: Artificial Intelligence and Statistics, 2007, pp. 548–555.

[59] G.W. Taylor, G.E. Hinton, S.T. Roweis, Modeling human motion using binary latent variables, in: Advances in Neural Information Processing Systems, 2007, pp. 1345–1352.

[60] R. Memisevic, G. Hinton, Unsupervised learning of image transformations, in: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–8.

[61] H. Lee, P. Pham, Y. Largman, A.Y. Ng, Unsupervised feature learning for audio classification using convolutional deep belief networks, in: Advances in Neural Information Processing Systems, 2009, pp. 1096–1104.

[62] G. Dahl, A.-r. Mohamed, G.E. Hinton, et al., Phone recognition with the mean-covariance restricted boltzmann machine, in: Advances in Neural Information Processing Systems, 2010, pp. 469–477.

[63] G.E. Hinton, et al., Modeling pixel means and covariances using factorized third-order boltzmann machines, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2551–2558.

[64] A.-r. Mohamed, G. Hinton, G. Penn, Understanding how deep belief networks perform acoustic modelling, in: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 4273–4276.

[65] I. Sutskever, G.E. Hinton, G.W. Taylor, The recurrent temporal restricted boltzmann machine, in: Advances in Neural Information Processing Systems, 2009, pp. 1601–1608.

[66] G.W. Taylor, G.E. Hinton, Factored conditional restricted boltzmann machines for modeling motion style, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 1025–1032.

[67] G.E. Hinton, A practical guide to training restricted boltzmann machines, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 599–619.

[68] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep learning, Vol. 1, MIT press Cambridge, 2016.

[69] N. Le Roux, Y. Bengio, Representational power of restricted boltzmann machines and deep belief networks, Neural Comput. 20 (6) (2008) 1631–1649.

[70] G.E. Hinton, et al., What kind of graphical model is the brain?, in: IJCAI, Vol. 5, 2005, pp. 1765–1775.

[71] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3626–3633.

[72] A.-r. Mohamed, G.E. Dahl, G. Hinton, et al., Acoustic modeling using deep belief networks, IEEE Trans. Audio Speech Lang. Process. 20 (1) (2012) 14–22.

[73] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, J. Mach. Learn. Res. 11 (Feb) (2010) 625–660.

[74] S.M. Siniscalchi, J. Li, C.-H. Lee, Hermitian polynomial for speaker adaptation of connectionist speech recognition systems, IEEE Trans. Audio Speech Lang. Process. 21 (10) (2013) 2152–2161.

[75] S.M. Siniscalchi, D. Yu, L. Deng, C.-H. Lee, Exploiting deep neural networks for detection-based speech recognition, Neurocomputing 106 (2013) 148–157.

[76] D. Yu, S.M. Siniscalchi, L. Deng, C.-H. Lee, Boosting attribute and phone estimation accuracies with deep neural networks for detection-based speech recognition, in: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 4169–4172.

[77] H. Lee, R. Grosse, R. Ranganath, A.Y. Ng, Unsupervised learning of hierarchical representations with convolutional deep belief networks, Commun. ACM 54 (10) (2011) 95–103.

[78] L. Zhao, Y. Zhou, H. Lu, H. Fujita, Parallel computing method of deep belief networks and its application to traffic flow prediction, Knowl.-Based Syst. 163 (2019) 972–987, http://dx.doi.org/10.1016/j.knosys.2018.10.025, URL http://www.sciencedirect.com/science/article/pii/S0950705118305112.

[79] R. Salakhutdinov, G. Hinton, Semantic hashing, Internat. J. Approx. Reason. 50 (7) (2009) 969–978.

[80] L. Deng, M.L. Seltzer, D. Yu, A. Acero, A.-r. Mohamed, G. Hinton, Binary coding of speech spectrograms using a deep auto-encoder, in: Eleventh Annual Conference of the International Speech Communication Association, 2010.

[81] C. Poultney, S. Chopra, Y.L. Cun, et al., Efficient learning of sparse representations with an energy-based model, in: Advances in Neural Information Processing Systems, 2007, pp. 1137–1144.

[82] L. Deng, The mnist database of handwritten digit images for machine learning research [best of the web], IEEE Signal Process. Mag. 29 (6) (2012) 141–142.

[83] J. Ngiam, Z. Chen, P.W. Koh, A.Y. Ng, Learning deep energy models, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 1105–1112.

[84] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 689–696.

[85] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[86] G. Alain, Y. Bengio, What regularized auto-encoders learn from the data-generating distribution, J. Mach. Learn. Res. 15 (1) (2014) 3563–3593.

[87] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[88] Y. Bengio, E. Laufer, G. Alain, J. Yosinski, Deep generative stochastic networks trainable by backprop, in: International Conference on Machine Learning, 2014, pp. 226–234.

[89] Y. Bengio, Deep learning of representations: Looking forward, in: International Conference on Statistical Language and Speech Processing, Springer, 2013, pp. 1–37.

[90] P. Vincent, A connection between score matching and denoising autoencoders, Neural Comput. 23 (7) (2011) 1661–1674.

[91] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (Dec) (2010) 3371–3408.

[92] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, 2012, arXiv preprint arXiv:1207.0580.

[93] C. Doersch, Tutorial on variational autoencoders, 2016, arXiv preprint arXiv:1606.05908.

[94] G.E. Hinton, A better way to learn features: technical perspective, Commun. ACM 54 (10) (2011) 94.

[95] G.E. Hinton, A. Krizhevsky, S.D. Wang, Transforming auto-encoders, in: International Conference on Artificial Neural Networks, Springer, 2011, pp. 44–51.

[96] Q.V. Le, Building high-level features using large scale unsupervised learning, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 8595–8598.

[97] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.

[98] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, in: NIPS'12, Curran Associates Inc., USA, 2012, pp. 1097–1105, URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

[99] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, URL http://www.deeplearningbook.org.

[100] Y. LeCun, Y. Bengio, G.E. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.

[101] C. Francois, Deep learning with Python, Manning Publications Company, 2017.

[102] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, S. Tubaro, Deep convolutional neural networks for pedestrian detection, Signal Process.: Image Commun. 47 (2016) 482–489.

[103] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. (2019).

[104] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[105] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.

[106] Y. Wang, Z. Wang, A survey of recent work on fine-grained image classification techniques, J. Vis. Commun. Image Represent. 59 (2019) 210–214.

[107] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2528–2535.

[108] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1520–1528.
[109] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[110] S. Fidler, R. Mottaghi, A. Yuille, R. Urtasun, Bottom-up segmentation for top-down detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3294–3301.
[111] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[112] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[113] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[114] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2147–2154.
[115] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[116] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.
[117] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.
[118] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, CoRR abs/1409.1556.
[119] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, http://dx.doi.org/10.1109/CVPR.2015.7298594.
[120] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, http://dx.doi.org/10.1109/CVPR.2016.90.
[121] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 818–833.
[122] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, 2018, CoRR abs/1808.03314.
[123] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.
[124] F.A. Gers, J. Schmidhuber, Recurrent nets that time and count, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 3, 2000, pp. 189–194, http://dx.doi.org/10.1109/IJCNN.2000.861302.
[125] J. Chung, Çaglar Gülçehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, CoRR abs/1412.3555.
[126] H. Sak, A.W. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: INTERSPEECH, 2014.
[127] P. Doetsch, M. Kozielski, H. Ney, Fast and robust training of recurrent neural networks for offline handwriting recognition, 2014 14th International Conference on Frontiers in Handwriting Recognition (2014) 279–284.
[128] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R.K. Ward, Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval, IEEE/ACM Trans. Audio Speech Lang. Process. 24 (2016) 694–707.
[129] M. Pota, F. Marulli, M. Esposito, G.D. Pietro, H. Fujita, Multilingual pos tagging by a composite deep architecture based on character-level features and on-the-fly enriched word embeddings, Knowl.-Based Syst. 164 (2019) 309–323, http://dx.doi.org/10.1016/j.knosys.2018.11.003, URL http://www.sciencedirect.com/science/article/pii/S0950705118305392.
[130] P. Gao, Q. Zhang, F. Wang, L. Xiao, H. Fujita, Y. Zhang, Learning reinforced attentional representation for end-to-end visual tracking, Inform. Sci. 517 (2020) 52–67, http://dx.doi.org/10.1016/j.ins.2019.12.084, URL http://www.sciencedirect.com/science/article/pii/S0020025519312095.
[131] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A.C. Courville, Y. Bengio, Generative adversarial nets, in: NIPS, 2014.
[132] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, in: NIPS'16, Curran Associates Inc., USA, 2016, pp. 2234–2242, URL http://dl.acm.org/citation.cfm?id=3157096.3157346.
[133] R. Groß, Y. Gu, W. Li, M. Gauci, Generalizing gans: A turing perspective, in: NIPS, 2017.
[134] F. Zhou, S. Yang, H. Fujita, D. Chen, C. Wen, Deep learning fault diagnosis method based on global optimization gan for unbalanced data, Knowl.-Based Syst. 187 (2020) 104837, http://dx.doi.org/10.1016/j.knosys.2019.07.008, URL http://www.sciencedirect.com/science/article/pii/S0950705119303120.
[135] J. Wu, C. Zhang, T. Xue, W.T. Freeman, J.B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling, in: NIPS, 2016.
[136] C. Vondrick, H. Pirsiavash, A. Torralba, Generating videos with scene dynamics, in: NIPS, 2016.
[137] S.E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, in: ICML, 2016.
[138] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 2818–2826.
[139] Y. Bengio, Learning deep architectures for ai, Found. Trends Mach. Learn. 2 (1) (2009) 1–127, http://dx.doi.org/10.1561/2200000006.
[140] S. Xie, R.B. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 5987–5995.
[141] G. Huang, Z. Liu, L. v. d. Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243.
[142] S. Woo, J. Park, J.-Y. Lee, I.-S. Kweon, Cbam: Convolutional block attention module, in: ECCV, 2018.
[143] Y. Hu, G. Wen, M. Luo, D. Dai, J. Ma, Competitive inner-imaging squeeze and excitation for residual network, 2018, ArXiv abs/1807.08920.
[144] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Ng, Reading digits in natural images with unsupervised feature learning, in: NIPS, 2011.
[145] K. Gregor, I. Danihelka, A. Graves, D.J. Rezende, D. Wierstra, Draw: A recurrent neural network for image generation, 2015, ArXiv abs/1502.04623.
[146] N. Kalchbrenner, I. Danihelka, A. Graves, Grid long short-term memory, 2015, CoRR abs/1507.01526.
[147] L. Jing, C. Gulcehre, J. Peurifoy, Y. Shen, M. Tegmark, M. Soljacic, Y. Bengio, Gated orthogonal recurrent units: On learning to forget, Neural Comput. 31 (4) (2019) 765–783, http://dx.doi.org/10.1162/neco_a_01174.
[148] F. Belletti, A. Beutel, S. Jain, E.H. hsin Chi, Factorized recurrent neural architectures for longer range dependence, in: AISTATS, 2018.
[149] F. Assunção, N. Lourenço, P. Machado, B. Ribeiro, Denser: deep evolutionary network structured representation, Genet. Program. Evol. Mach. (2018) 1–31.
[150] B.A. Garro, R.A. Vázquez, Designing artificial neural networks using particle swarm optimization algorithms, in: Comp. Int. and Neurosc., 2015.
[151] G. Das, P.K. Pattnaik, S.K. Padhy, Artificial neural network trained by particle swarm optimization for non-linear channel equalization, Expert Syst. Appl. 41 (7) (2014) 3491–3496, http://dx.doi.org/10.1016/j.eswa.2013.10.053, URL http://www.sciencedirect.com/science/article/pii/S0957417413008701.
[152] B. Wang, Y. Sun, B. Xue, M. Zhang, Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification, in: 2018 IEEE Congress on Evolutionary Computation (CEC), 2018, pp. 1–8.
[153] S. Sengupta, S. Basak, R.A. Peters, Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives, Mach. Learn. Knowl. Extraction 1 (1) (2018) 157–191, http://dx.doi.org/10.3390/make1010010, URL http://www.mdpi.com/2504-4990/1/1/10.
[154] S. Sengupta, S. Basak, R.A. Peters, Qdds: A novel quantum swarm algorithm inspired by a double dirac delta potential, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 704–711, http://dx.doi.org/10.1109/SSCI.2018.8628792.
[155] S. Sengupta, S. Basak, R.A. Peters, Chaotic quantum double delta swarm algorithm using chebyshev maps: theoretical foundations, performance analyses and convergence issues, J. Sensor Actuator Netw. 8 (1) (2019) http://dx.doi.org/10.3390/jsan8010009, URL http://www.mdpi.com/2224-2708/8/1/9.
[156] B. Dhariyal, V. Ravi, Word2vec and evolutionary computing driven hybrid deep learning based sentiment analysis, in: Proceedings of International Conference on Soft Computing and Signal Processing, ICSCSP 2019, Springer International Publishing, 2020, (in press).
[157] M. Hüttenrauch, A. Sosic, G. Neumann, Deep reinforcement learning for swarm systems, 2018, CoRR abs/1807.06613.
[158] C. Anderson, A.V. Mayrhauser, R. Mraz, On the use of neural networks to guide software testing activities, in: Proceedings of 1995 IEEE International Test Conference (ITC), 1995, pp. 720–729, http://dx.doi.org/10.1109/TEST.1995.529902.

[159] T.M. Khoshgoftaar, R.M. Szabo, Using neural networks to predict software faults during testing, IEEE Trans. Reliab. 45 (3) (1996) 456–462, http://dx.doi.org/10.1109/24.537016.

[160] M. Vanmali, M. Last, A. Kandel, Using a neural network in the software testing process, Int. J. Intell. Syst. 17 (2002) 45–62.

[161] Y. Sun, X. Huang, D. Kroening, Testing deep neural networks, 2018, CoRR abs/1803.04792.

[162] G. Katz, C.W. Barrett, D.L. Dill, K. Julian, M.J. Kochenderfer, Towards proving the adversarial robustness of deep neural networks., in: FVAV@IFM, 2017.

[163] X. Huang, M.Z. Kwiatkowska, S. Wang, M. Wu, Safety verification of deep neural networks, in: CAV, 2017.

[164] C.E. Tuncali, G. Fainekos, H. Ito, J. Kapinski, Simulation-based adversarial test generation for autonomous vehicles with machine learning components, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018, pp. 1555–1562.

[165] X. Yuan, P. He, Q. Zhu, R.R. Bhat, X. Li, Adversarial examples: Attacks and defenses for deep learning, 2017, CoRR abs/1712.07107.

[166] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, 2014, CoRR abs/1412.6572.

[167] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: A simple and accurate method to fool deep neural networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2574–2582.

[168] B.D. Rouhani, M. Samragh, M. Javaheripi, T. Javidi, F. Koushanfar, Deepfense: Online accelerated defense against adversarial deep learning, in: Proceedings of the International Conference on Computer-Aided Design, in: ICCAD '18, ACM, New York, NY, USA, 2018, pp. 134:1–134:8, http://dx.doi.org/10.1145/3240765.3240791, URL http://doi.acm.org/10.1145/3240765.3240791.

[169] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, D. Mukhopadhyay, Adversarial attacks and defences: A survey, 2018, CoRR abs/1810.00069.

[170] A. Pumsirirat, L. Yan, Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine, Int. J. Adv. Comput. Sci. Appl. 9 (1) (2018) 18–25.

[171] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, B. Reimer, Detection of anomalies in large scale accounting data using deep autoencoder networks, 2017, CoRR abs/1709.05254, arXiv:1709.05254. URL http://arxiv.org/abs/1709.05254.

[172] Y. Wang, W. Xu, Leveraging deep learning with lda-based text analytics to detect automobile insurance fraud, Decis. Support Syst. 105 (2018) 87–95.

[173] A.K. Gangwar, V. Ravi, Wip: Generative adversarial network for oversampling data in credit card fraud detection, in: D. Garg, N.V.N. Kumar, R.K. Shyamasundar (Eds.), Information Systems Security, Springer International Publishing, Cham, 2019, pp. 123–134.

[174] Y.-J. Zheng, X.-H. Zhou, W.-G. Sheng, Y. Xue, S.-Y. Chen, Generative adversarial network based telecom fraud detection at the receiving bank, Neural Netw. 102 (2018) 78–86, http://dx.doi.org/10.1016/j.neunet.2018.02.015, URL http://www.sciencedirect.com/science/article/pii/S0893608018300698.

[175] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, X. Ning, Opinion fraud detection via neural autoencoder decision forest, Pattern Recognit. Lett. (2018) http://dx.doi.org/10.1016/j.patrec.2018.07.013, URL http://www.sciencedirect.com/science/article/pii/S0167865518303039.

[176] J.A. Gómez, J. Arévalo, R. Paredes, J. Nin, End-to-end neural network architecture for fraud scoring in card payments, Pattern Recognit. Lett. 105 (2018) 175–181, http://dx.doi.org/10.1016/j.patrec.2017.08.024, URL http://www.sciencedirect.com/science/article/pii/S016786551730291X, Machine learning and applications in artificial intelligence.

[177] N.F. Ryman-Tubb, P. Krause, W. Garn, How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark, Eng. Appl. Artif. Intell. 76 (2018) 130–157, http://dx.doi.org/10.1016/j.engappai.2018.07.008, URL http://www.sciencedirect.com/science/article/pii/S0952197618301520.

[178] U. Fiore, A.D. Santis, F. Perla, P. Zanetti, F. Palmieri, Using generative adversarial networks for improving classification effectiveness in credit card fraud detection, Inform. Sci. 479 (2019) 448–455, http://dx.doi.org/10.1016/j.ins.2017.12.030, URL http://www.sciencedirect.com/science/article/pii/S0020025517311519.

[179] R.C. Cavalcante, R.C. Brasileiro, V.L. Souza, J.P. Nobrega, A.L. Oliveira, Computational intelligence and financial markets: A survey and future directions, Expert Syst. Appl. 55 (2016) 194–211.

[180] X. Li, Z. Deng, J. Luo, Trading strategy design in financial investment through a turning points prediction scheme, Expert Syst. Appl. 36 (4) (2009) 7818–7826, http://dx.doi.org/10.1016/j.eswa.2008.11.014, URL http://www.sciencedirect.com/science/article/pii/S0957417408008622.

[181] E.F. Fama, Random walks in stock market prices, Financ. Anal. J. 51 (1) (1995) 75–80.

[182] C.-J. Lu, T.-S. Lee, C.-C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decis. Support Syst. 47 (2) (2009) 115–125, http://dx.doi.org/10.1016/j.dss.2009.02.001, URL http://www.sciencedirect.com/science/article/pii/S0167923609000323.

[183] M. Tkáč, R. Verner, Artificial neural networks in business: Two decades of research, Appl. Soft Comput. 38 (2016) 788–804, http://dx.doi.org/10.1016/j.asoc.2015.09.040, URL http://www.sciencedirect.com/science/article/pii/S1568494615006122.

[184] T.N. Pandey, A.K. Jagadev, S. Dehuri, S.-B. Cho, A novel committee machine and reviews of neural network and statistical models for currency exchange rate prediction: An experimental analysis, J. King Saud Univ. - Comput. Inf. Sci. (2018) http://dx.doi.org/10.1016/j.jksuci.2018.02.010, URL http://www.sciencedirect.com/science/article/pii/S1319157817303816.

[185] A. Lasfer, H. El-Baz, I. Zualkernan, Neural network design parameters for forecasting financial time series, in: Modeling, Simulation and Applied Optimization (ICMSAO), 2013 5th International Conference on, IEEE, 2013, pp. 1–4.

[186] M.U. Gudelek, S.A. Boluk, A.M. Ozbayoglu, A deep learning based stock trading model with 2-d cnn trend detection, in: Computational Intelligence (SSCI), 2017 IEEE Symposium Series on, IEEE, 2017, pp. 1–8.

[187] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, European J. Oper. Res. 270 (2) (2018) 654–669.

[188] L. dos Santos Pinheiro, M. Dras, Stock market prediction with deep learning: A character-based neural language model for event-based trading, in: Proceedings of the Australasian Language Technology Association Workshop 2017, 2017, pp. 6–15.

[189] W. Bao, J. Yue, Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory, PLoS One 12 (7) (2017) e0180944.

[190] A.H. Mohammad, K. Rezaul, T. Ruppa, D.B.B. Neil, W. Yang, Hybrid deep learning model for stock price prediction, in: IEEE Symposium Symposium Series on Computational Intelligence SSCI, 2018, IEEE, 2018, pp. 1837–1844.

[191] A. le Calvez, D. Cliff, Deep learning can replicate adaptive traders in a limit-order-book financial market, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1876–1883.

[192] S. Basak, S. Sengupta, A. Dubey, Mechanisms for integrated feature normalization and remaining useful life estimation using LSTMs applied to hard-disks, 2018, arXiv e-prints arXiv:1810.08985.

[193] P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, Reliab. Eng. Syst. Saf. 115 (2013) 124–135, http://dx.doi.org/10.1016/j.ress.2013.02.022, URL http://www.sciencedirect.com/science/article/pii/S0951832013000574.

[194] J. Sikora, Disk failures in the real world : What does an mttf of 1 , 000 , 000 hours mean to you ?, 2007.

[195] G. Wang, L. Zhang, W. Xu, What Can we learn from four years of data center hardware failures?, in: 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017, pp. 25–36, http://dx.doi.org/10.1109/DSN.2017.26.

[196] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using a deep belief network with restricted boltzmann machines, Neurocomputing 137 (2014) 47–56, http://dx.doi.org/10.1016/j.neucom.2013.03.047, URL http://www.sciencedirect.com/science/article/pii/S0925231213007388, Advanced intelligent computing theories and methodologies.

[197] J. Qiu, W. Liang, L. Zhang, X. Yu, M. Zhang, The early-warning model of equipment chain in gas pipeline based on dnn-hmm, J. Natural Gas Sci. Eng. 27 (2015) 1710–1722, http://dx.doi.org/10.1016/j.jngse.2015.10.036, URL http://www.sciencedirect.com/science/article/pii/S1875510015302407.

[198] N. Gugulothu, T. Vishnu, P. Malhotra, L. Vig, P. Agarwal, G. Shroff, Predicting remaining useful life using time series embeddings based on recurrent neural networks, 2017, CoRR.

[199] P. Filonov, A. Lavrentyev, A. Vorontsov, Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model, 2016, CoRR abs/1612.06676.

[200] M.M. Botezatu, I. Giurgiu, J. Bogojeska, D. Wiesmann, Predicting disk replacement towards reliable data centers, in: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '16, ACM, New York, NY, USA, 2016, pp. 39–48, http://dx.doi.org/10.1145/2939672.2939699, URL http://doi.acm.org/10.1145/2939672.2939699.

[201] W. Fei, G. Song, J. Zang, Y. Gao, J. Sun, L. Yu, Framework model for time-variant propagation speed and congestion boundary by incident on expressways, IET Intell. Transp. Syst. 11 (1) (2017) 10–17, http://dx.doi.org/10.1049/iet-its.2015.0222.

[202] S.L. Zhang, Y.Z. Yao, J. Hu, Y. Zhao, S. Li, J. Hu, Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks, in: Sensors, 2019.

[203] X. Ma, H. Yu, Y. Wang, Y. Wang, Large-scale transportation network congestion evolution prediction using deep learning theory, PLoS One 10 (2015) e0119044, http://dx.doi.org/10.1371/journal.pone.0119044.

[204] H.-I. Suk, S.-W. Lee, D. Shen, Latent feature representation with stacked auto-encoder for ad/mci diagnosis, Brain Struct. Funct. 220 (2013) 841–859.

[205] G. van Tulder, M. de Bruijne, Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines, IEEE Trans. Med. Imaging 35 (5) (2016) 1262–1272, http://dx.doi.org/10.1109/TMI.2016.2526687.

[206] T. Brosch, R. Tam, Manifold learning of brain mris by deep learning, in: K. Mori, I. Sakuma, Y. Sato, C. Barillot, N. Navab (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 633–640.

[207] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, Nature 542 (2017) 115–.

[208] S. Rajaraman, S.K. Antani, M. Poostchi, K. Silamut, M.A. Hossain, R.J. Maude, S. Jaeger, G.R. Thoma, Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images, PeerJ 6 (2018) e4568, http://dx.doi.org/10.7717/peerj.4568, URL https://www.ncbi.nlm.nih.gov/pubmed/29682411 29682411[pmid].

[209] G. Kang, K. Liu, B. Hou, N. Zhang, 3d multi-view convolutional neural networks for lung nodule classification, in: PloS One, 2017.

[210] S. Hwang, H. Kim, Self-transfer learning for fully weakly supervised object localization, 2016, CoRR abs/1602.01625. URL http://arxiv.org/abs/1602.01625.

[211] S. Andermatt, S. Pezold, P. Cattin, Multi-dimensional gated recurrent units for the segmentation of biomedical 3d-data, in: G. Carneiro, D. Mateus, L. Peter, A. Bradley, J.M.R.S. Tavares, V. Belagiannis, J.P. Papa, J.C. Nascimento, M. Loog, Z. Lu, J.S. Cardoso, J. Cornebise (Eds.), Deep Learning and Data Labeling for Medical Applications, Springer International Publishing, Cham, 2016, pp. 142–151.

[212] X. Cheng, X. Lin, Y. Zheng, Deep similarity learning for multimodal medical images, CMBBE: Imaging Vis. 6 (2018) 248–252.

[213] S. Miao, Z.J. Wang, R. Liao, A cnn regression approach for real-time 2d/3d registration, IEEE Trans. Med. Imaging 35 (5) (2016) 1352–1363, http://dx.doi.org/10.1109/TMI.2016.2521800.

[214] O. Oktay, W. Bai, M.C.H. Lee, R. Guerrero, K. Kamnitsas, J. Caballero, A. de Marvao, S.A. Cook, D.P. O'Regan, D. Rueckert, Multi-input cardiac image super-resolution using convolutional neural networks, in: MICCAI, 2016.

[215] V. Golkov, A. Dosovitskiy, J.I. Sperl, M.I. Menzel, M. Czisch, P. Sämann, T. Brox, D. Cremers, Q-space deep learning: Twelve-fold shorter and model-free diffusion mri scans, IEEE Trans. Med. Imaging 35 (5) (2016) 1344–1351, http://dx.doi.org/10.1109/TMI.2016.2551324.

[216] G.J.S. Litjens, T. Kooi, B.E. Bejnordi, A.A.A. Setio, F. Ciompi, M. Ghafoorian, J.A.W.M. van der Laak, B. van Ginneken, C.I. Sánchez, A survey on deep learning in medical image analysis, 2017, CoRR abs/1702.05747. URL http://arxiv.org/abs/1702.05747.

[217] O. Yildirim, R.S. Tan, U.R. Acharya, An efficient compression of ecg signals using deep convolutional autoencoders, Cogn. Syst. Res. 52 (2018) 198–211, http://dx.doi.org/10.1016/j.cogsys.2018.07.004, URL http://www.sciencedirect.com/science/article/pii/S1389041718302730.

[218] A. Gangwar, V. Ravi, Diabetic retinopathy detection using transfer learning and deep learning, in: Proceedings of the 8th International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA 2020), in: Advances in Intelligent Systems and Computing, Springer International Publishing, 2020, (in press).

[219] O. Yildirim, M. Talo, B. Ay, U.B. Baloglu, G. Aydin, U.R. Acharya, Automated detection of diabetic subject using pre-trained 2d-cnn models with frequency spectrum images extracted from heart rate signals, Comput. Biol. Med. 113 (2019) 103387, http://dx.doi.org/10.1016/j.compbiomed.2019.103387, URL http://www.sciencedirect.com/science/article/pii/S0010482519302641.

[220] U. Raghavendra, H. Fujita, S.V. Bhandary, A. Gudigar, J.H. Tan, U.R. Acharya, Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images, Inform. Sci. 441 (2018) 41–49, http://dx.doi.org/10.1016/j.ins.2018.01.051, URL http://www.sciencedirect.com/science/article/pii/S0020025518300744.

[221] M. Talo, O. Yildirim, U.B. Baloglu, G. Aydin, U.R. Acharya, Convolutional neural networks for multi-class brain disease detection using mri images, Comput. Med. Imaging Graph. (2019) 101673, http://dx.doi.org/10.1016/j.compmedimag.2019.101673, URL http://www.sciencedirect.com/science/article/pii/S0895611119300886.

[222] U.B. Baloglu, M. Talo, O. Yildirim, R.S. Tan, U.R. Acharya, Classification of myocardial infarction with multi-lead ecg signals and deep cnn, Pattern Recognit. Lett. 122 (2019) 23–30, http://dx.doi.org/10.1016/j.patrec.2019.02.016, URL http://www.sciencedirect.com/science/article/pii/S016786551930056X.

[223] M. Talo, U.B. Baloglu, Özal Yıldırım, U.R. Acharya, Application of deep transfer learning for automated brain abnormality classification using mr images, Cogn. Syst. Res. 54 (2019) 176–188, http://dx.doi.org/10.1016/j.cogsys.2018.12.007, URL http://www.sciencedirect.com/science/article/pii/S1389041718310933.

[224] V.S.S. Vankayala, N.D. Rao, Artificial neural networks and their applications to power systems—a bibliographical survey, Electr. Power Syst. Res. 28 (1) (1993) 67–79.

[225] M.-y. Chow, P. Mangum, R. Thomas, Incipient fault detection in dc machines using a neural network, in: Twenty-Second Asilomar Conference on Signals, Systems and Computers, 1988, Vol. 2, IEEE, 1988, pp. 706–709.

[226] Z. Guo, K. Zhou, X. Zhang, S. Yang, A deep learning model for short-term power load and probability density forecasting, Energy 160 (2018) 1186–1200.

[227] R.E. Bourguet, P.J. Antsaklis, Artificial neural networks in electric power industry, ISIS 94 (1994) 007.

[228] J. Sharp, Comparative models for electrical load forecasting: D.H. Bunn and E.D. Farmer, eds.(wiley, new york, 1985) [uk pound]24.95, pp. 232, Int. J. Forecast. 2 (2) (1986) 241–242, URL https://EconPapers.repec.org/RePEc:eee:intfor:v:2:y:1986:i:2:p:241-242.

[229] H.S. Hippert, C.E. Pedreira, R.C. Souza, Neural networks for short-term load forecasting: A review and evaluation, IEEE Trans. Power Syst. 16 (1) (2001) 44–55.

[230] C. Kuster, Y. Rezgui, M. Mourshed, Electrical load forecasting models: A critical systematic review, Sustain. Cities Soc. 35 (2017) 257–270.

[231] R. Aggarwal, Y. Song, Artificial neural networks in power systems. i. general introduction to neural computing, Power Eng. J. 11 (3) (1997) 129–134.

[232] Y. Zhai, Time Series Forecasting Competition among Three Sophisticated Paradigms (Ph.D. thesis), University of North Carolina at Wilmington, 2005.

[233] D.C. Park, M. El-Sharkawi, R. Marks, L. Atlas, M. Damborg, Electric load forecasting using an artificial neural network, IEEE Trans. Power Syst. 6 (2) (1991) 442–449.

[234] E. Mocanu, P.H. Nguyen, M. Gibescu, W.L. Kling, Deep learning for estimating building energy consumption, Sustain. Energy Grids Netw. 6 (2016) 91–99.

[235] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, J. He, Short-term load forecasting with deep residual networks, IEEE Trans. Smart Grid (2018).

[236] S. Bouktif, A. Fiaz, A. Ouni, M. Serhani, Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches, Energies 11 (7) (2018) 1636.

[237] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[238] S. Li, L. Goel, P. Wang, An ensemble approach for short-term load forecasting by extreme learning machine, Appl. Energy 170 (2016) 22–29.

[239] C. Cecati, J. Kolbusz, P. Różycki, P. Siano, B.M. Wilamowski, A novel rbf training algorithm for short-term electric load forecasting and comparative studies, IEEE Trans. Ind. Electron. 62 (10) (2015) 6519–6529.

[240] A. Dedinec, S. Filiposka, A. Dedinec, L. Kocarev, Deep belief network based electricity load forecasting: An analysis of macedonian case, Energy 115 (2016) 1688–1700.

[241] A. Rahman, V. Srikumar, A.D. Smith, Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks, Appl. Energy 212 (2018) 372–385.

[242] W. Kong, Z.Y. Dong, Y. Jia, D.J. Hill, Y. Xu, Y. Zhang, Short-term residential load forecasting based on lstm recurrent neural network, IEEE Trans. Smart Grid (2017).

[243] X. Dong, L. Qian, L. Huang, Short-term load forecasting in smart grid: A combined cnn and k-means clustering approach, in: 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), IEEE, 2017, pp. 119–125.

[244] S.A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renew. Sustain. Energy Rev. 5 (4) (2001) 373–401.

[245] H. Wang, H. Yi, J. Peng, G. Wang, Y. Liu, H. Jiang, W. Liu, Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network, Energy Convers. Manage. 153 (2017) 409–422.

[246] U.K. Das, K.S. Tey, M. Seyedmahmoudian, S. Mekhilef, M.Y.I. Idris, W. Van Deventer, B. Horan, A. Stojcevski, Forecasting of photovoltaic power generation and model optimization: A review, Renew. Sustain. Energy Rev. 81 (2018) 912–928.

[247] V. Dabra, K.K. Paliwal, P. Sharma, N. Kumar, Optimization of photovoltaic power system: a comparative study, Prot. Control Mod. Power Syst. 2 (1) (2017) 3.

[248] J. Liu, W. Fang, X. Zhang, C. Yang, An improved photovoltaic power forecasting model with the assistance of aerosol index data, IEEE Trans. Sustain. Energy 6 (2) (2015) 434–442.

[249] H.S. Jang, K.Y. Bae, H.-S. Park, D.K. Sung, Solar power prediction based on satellite images and support vector machine, IEEE Trans. Sustain. Energy 7 (3) (2016) 1255–1263.

[250] A. Gensler, J. Henze, B. Sick, N. Raabe, Deep learning for solar power forecasting—An approach using autoencoder and lstm neural networks, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2016, pp. 002858–002865.

[251] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep lstm-rnn, Neural Comput. Appl. (2017) 1–14.

[252] J.F. Manwell, J.G. McGowan, A.L. Rogers, Wind Eenergy Explained: Theory, Design and Application, John Wiley & Sons, 2010.

[253] A.P. Marugán, F.P.G. Márquez, J.M.P. Perez, D. Ruiz-Hernández, A survey of artificial neural network in wind energy systems, Appl. Energy 228 (2018) 1822–1836.

[254] W. Wu, K. Chen, Y. Qiao, Z. Lu, Probabilistic short-term wind power forecasting based on deep neural networks, in: 2016 International Conference on Probabilistic Methods Applied To Power Systems (PMAPS), IEEE, 2016, pp. 1–8.

[255] H.-z. Wang, G.-q. Li, G.-b. Wang, J.-c. Peng, H. Jiang, Y.-t. Liu, Deep learning based ensemble approach for probabilistic wind power forecasting, Appl. Energy 188 (2017) 56–70.

[256] K. Wang, X. Qi, H. Liu, J. Song, Deep belief network based k-means cluster approach for short-term wind power forecasting, Energy 165 (2018) 840–852.

[257] A.S. Qureshi, A. Khan, A. Zameer, A. Usman, Wind power prediction using deep neural network based meta regression and transfer learning, Appl. Soft Comput. 58 (2017) 742–755.

[258] N. Marz, J. Warren, Big Data: Principles and Best Practices of Scalable Real-Time Data Systems, Manning Publications Co., New York, 2015.

[259] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. (6) (2005) 734–749.

[260] D. Bokde, S. Girase, D. Mukhopadhyay, Matrix factorization model in collaborative filtering algorithms: A survey, Procedia Comput. Sci. 49 (2015) 136–146.

[261] S. Sedhain, A.K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th International Conference on World Wide Web, ACM, 2015, pp. 111–112.

[262] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, J. Zhu, Personal recommendation using deep recurrent neural networks in netease, in: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 1218–1229.

[263] H. Wang, N. Wang, D.-Y. Yeung, Collaborative deep learning for recommender systems, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1235–1244.

[264] K. Georgiev, P. Nakov, A non-iid framework for collaborative filtering with restricted boltzmann machines, in: International Conference on Machine Learning, 2013, pp. 1148–1156.

[265] X. Liu, Y. Ouyang, W. Rong, Z. Xiong, Item category aware conditional restricted boltzmann machine based recommendation, in: International Conference on Neural Information Processing, Springer, 2015, pp. 609–616.

[266] C. Hongliang, Q. Xiaona, The video recommendation system based on dbn, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 1016–1021.

[267] X. Wang, Y. Wang, Improving content-based and hybrid music recommendation using deep learning, in: Proceedings of the 22nd ACM International Conference on Multimedia, ACM, 2014, pp. 627–636.

[268] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Advances in Neural Information Processing Systems, 2013, pp. 2643–2651.

[269] L. Zheng, V. Noroozi, P.S. Yu, Joint deep modeling of users and items using reviews for recommendation, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 425–434.

[270] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, ACM, 2016, pp. 233–240.

[271] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.

[272] Y. Tay, A.T. Luu, S.C. Hui, Multi-pointer co-attention networks for recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2309–2318.

[273] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[274] M.D. Zeiler, Adadelta: an adaptive learning rate method, 2012, arXiv preprint arXiv:1212.5701.

[275] S. Basak, S. Sengupta, A. Dubey, Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks, in: 2019 IEEE International Conference on Smart Computing (SMARTCOMP), 2019, pp. 208–216, http://dx.doi.org/10.1109/SMARTCOMP.2019.00055.

[276] [link]. URL https://www.backblaze.com/blog/hard-drive-smart-stats/.

[277] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (1) (2017) 221–248, http://dx.doi.org/10.1146/annurev-bioeng-071516-044442, pMID: 28301734. arXiv:https://doi.org/10.1146/annurev-bioeng-071516-044442.

[278] A. Işın, C. Direkoğlu, M. Şah, Review of MRI-based brain tumor image segmentation using deep learning methods, Procedia Comput. Sci. 102 (2016) 317–324, http://dx.doi.org/10.1016/j.procs.2016.09.407, 12th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria. URL http://www.sciencedirect.com/science/article/pii/S187705091632587X.

[279] B.H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. Weber, T. Arbel, B.B. Avants, N. Ayache, P. Buendia, D.L. Collins, N. Cordier, J.J. Corso, A. Criminisi, T. Das, H. Delingette, Demiralp, C.R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K.M. Iftekharuddin, R. Jena, N.M. John, E. Konukoglu, D. Lashkari, J.A. Mariz, R. Meier, S. Pereira, D. Precup, S.J. Price, T.R. Raviv, S.M.S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. Shin, J. Shotton, C.A. Silva, N. Sousa, N.K. Subbanna, G. Szekely, T.J. Taylor, O.M. Thomas, N.J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D.H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, K. Van Leemput, The multimodal brain tumor image segmentation benchmark (brats), IEEE Trans. Med. Imaging 34 (10) (2015) 1993–2024, http://dx.doi.org/10.1109/TMI.2014.2377694.

[280] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[281] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, Robust visual tracking via multi-task sparse learning, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2042–2049, http://dx.doi.org/10.1109/CVPR.2012.6247908.

[282] U. Raghavendra, A. Gudigar, S.V. Bhandary, T.N. Rao, E.J. Ciaccio, U.R. Acharya, A two layer sparse autoencoder for glaucoma identification with fundus images, J. Med. Syst. 43 (9) (2019) 299, http://dx.doi.org/10.1007/s10916-019-1427-x.

[283] H. Fujita, D. Cimr, Computer aided detection for fibrillations and flutters using deep convolutional neural network, Inform. Sci. 486 (2019) 231–239, http://dx.doi.org/10.1016/j.ins.2019.02.065, URL http://www.sciencedirect.com/science/article/pii/S0020025519301884.

[284] U.R. Acharya, H. Fujita, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals, Inform. Sci. 415–416 (2017) 190–198, http://dx.doi.org/10.1016/j.ins.2017.06.027, URL http://www.sciencedirect.com/science/article/pii/S0020025517308009.

[285] U.R. Acharya, H. Fujita, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, R.S. Tan, Deep convolutional neural network for the automated diagnosis of congestive heart failure using ecg signals, Appl. Intell. 49 (1) (2019) 16–27, http://dx.doi.org/10.1007/s10489-018-1179-1.

[286] H. Fujita, D. Cimr, Decision support system for arrhythmia prediction using convolutional neural network structure without preprocessing, Appl. Intell. 49 (9) (2019) 3383–3391, http://dx.doi.org/10.1007/s10489-019-01461-0.

[287] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, 2015, CoRR abs/1505.04597. arXiv:1505.04597. URL http://arxiv.org/abs/1505.04597.

[288] A. Maier, C. Syben, T. Lasser, C. Riess, A gentle introduction to deep learning in medical image processing, Z. Med. Phys. 29 (2) (2019) 89–101, http://dx.doi.org/10.1016/j.zemedi.2018.12.003, URL http://www.sciencedirect.com/science/article/pii/S093938891830120X.

[289] G. Grassi, P. Vecchio, Wind energy prediction using a two-hidden layer neural network, Commun. Nonlinear Sci. Numer. Simul. 15 (9) (2010) 2262–2266.

[290] N. Amjady, F. Keynia, H. Zareipour, Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization, IEEE Trans. Sustain. Energy 2 (3) (2011) 265–276.

[291] A. Zameer, J. Arshad, A. Khan, M.A.Z. Raja, Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks, Energy Convers. Manage. 134 (2017) 361–372.
[292] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.
[293] A. Voulodimos, N.D. Doulamis, A.D. Doulamis, E. Protopapadakis, Deep learning for computer vision: A brief review, in: Comp. Int. and Neurosc., 2018.
[294] A. Kamilaris, F.X. Prenafeta-Boldú, Deep learning in agriculture: A survey, Comput. Electron. Agric. 147 (2018) 70–90, http://dx.doi.org/10.1016/j.compag.2018.02.016, URL http://www.sciencedirect.com/science/article/pii/S0168169917308803.
[295] Q. Zhang, L.T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Inf. Fusion 42 (2018) 146–157, http://dx.doi.org/10.1016/j.inffus.2017.10.006, URL http://www.sciencedirect.com/science/article/pii/S1566253517305328.
[296] J.R. del Solar, P. Loncomilla, N. Soto, A survey on deep learning methods for robot vision, 2018, CoRR abs/1803.10862.
[297] A. Ioannidou, E. Chatzilari, S. Nikolopoulos, I. Kompatsiaris, Deep learning advances in computer vision with 3d data: A survey, ACM Comput. Surv. 50 (2017) http://dx.doi.org/10.1145/3042064.
[298] C. Seifert, A. Aamir, A. Balagopalan, D. Jain, A. Sharma, S. Grottel, S. Gumhold, Visualizations of deep neural networks in computer vision: A survey, in: T. Cerquitelli, D. Quercia, F. Pasquale (Eds.), Transparent Data Mining for Big and Small Data, Springer International Publishing, Cham, 2017, pp. 123–144, http://dx.doi.org/10.1007/978-3-319-54024-5_6, URL https://doi.org/10.1007/978-3-319-54024-5_6.
[299] W.-Y. Lin, Y.-H. Hu, C.-F. Tsai, Machine learning in financial crisis prediction: A survey, IEEE Trans. Syst. Man Cybern. - TSMC 42 (2012) 421–436, http://dx.doi.org/10.1109/TSMCC.2011.2170420.
[300] J.C.B. Gamboa, Deep learning for time-series analysis, 2017, CoRR abs/1701.01887.
[301] S. Sarojini Devi, Y. Radhika, A survey on machine learning and statistical techniques in bankruptcy prediction, Int. J. Mach. Learn. Comput. 8 (2018) 133–139, http://dx.doi.org/10.18178/ijmlc.2018.8.2.676.
[302] A. Tealab, Time series forecasting using artificial neural networks methodologies: a systematic review, Future Comput. Inf. J. 3 (2) (2018) 334–340, http://dx.doi.org/10.1016/j.fcij.2018.10.003, URL http://www.sciencedirect.com/science/article/pii/S2314728817300715.
[303] A. Almalaq, G. Edwards, A review of deep learning methods applied on load forecasting, in: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 511–516, http://dx.doi.org/10.1109/ICMLA.2017.0-110.
[304] J.P. Usuga Cadavid, S. Lamouri, B. Grabot, Trends in machine learning applied to demand & sales forecasting: A review, in: International Conference on Information Systems, Logistics and Supply Chain, Lyon, France, 2018. URL https://hal.archives-ouvertes.fr/hal-01881362.
[305] S. Beheshti-Kashi, H. Reza Karimi, K.-D. Thoben, M. Lütjen, M. Teucke, A survey on retail sales forecasting and prediction in fashion markets, Syst. Sci. Control Eng. Open Access J. 3 (2015) 154–161, http://dx.doi.org/10.1080/21642583.2014.999389.
[306] H.K. Alfares, M. Nazeeruddin, Electric load forecasting: Literature survey and classification of methods, Int. J. Syst. Sci. 33 (2002) 23–34.
[307] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24, http://dx.doi.org/10.1016/j.patrec.2014.01.008, URL http://www.sciencedirect.com/science/article/pii/S0167865514000221.
[308] M.Z. Hossain, F. Sohel, M.F. Shiratuddin, H. Laga, A comprehensive survey of deep learning for image captioning, ACM Comput. Surv. 51 (6) (2019) 118:1–118:36, http://dx.doi.org/10.1145/3295748, URL http://doi.acm.org/10.1145/3295748.
[309] H. Wang, S. Shang, L. Long, R. Hu, Y. Wu, N. Chen, S. Zhang, F. Cong, S. Lin, Biological image analysis using deep learning-based methods: literature review, Digit. Med. 4 (4) (2018) 157–165, http://dx.doi.org/10.4103/digm.digm_16_18, arXiv:http://www.digitmedicine.com/article.asp?issn=2226-8561;year=2018;volume=4;issue=4;spage=157;epage=165;aulast=Wang;t=6. URL http://www.digitmedicine.com/article.asp?issn=2226-8561;year=2018;volume=4;issue=4;spage=157;epage=165;aulast=Wang;t=6.
[310] Y. Li, H. Zhang, X. Xue, Y. Jiang, Q. Shen, Deep learning for remote sensing image classification: A survey, Wiley Interdiscip. Rev. Data Min. Knowl. Discovery 8 (6) (2018) e1264, http://dx.doi.org/10.1002/widm.1264, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1264. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1264.
[311] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural Comput. 29 (9) (2017) 2352–2449, http://dx.doi.org/10.1162/neco_a_00990, pMID: 28599112. arXiv:https://doi.org/10.1162/neco_a_00990. URL https://doi.org/10.1162/neco_a_00990.
[312] M.I. Razzak, S. Naz, A. Zaib, Deep learning for medical image processing: Overview, challenges and the future, in: N. Dey, A.S. Ashour, S. Borra (Eds.), Classification in BioApps: Automation of Decision Making, Springer International Publishing, Cham, 2018, pp. 323–350, http://dx.doi.org/10.1007/978-3-319-65981-7_12.
[313] A. Lundervold, A. Lundervold, An overview of deep learning in medical imaging focusing on mri, Z. Med. Phys. 29 (2) (2019) 102–127, http://dx.doi.org/10.1016/j.zemedi.2018.11.002, URL http://www.sciencedirect.com/science/article/pii/S0939388918301181.
[314] S. Liu, Y. Wang, X. Yang, B. Lei, L. Liu, S.X. Li, D. Ni, T. Wang, Deep learning in medical ultrasound analysis: A review, Engineering 5 (2) (2019) 261–275, http://dx.doi.org/10.1016/j.eng.2018.11.020, URL http://www.sciencedirect.com/science/article/pii/S2095809918301887.
[315] N. Akhtar, A.S. Mian, Threat of adversarial attacks on deep learning in computer vision: A survey, IEEE Access 6 (2018) 14410–14430.
[316] D.J. Miller, Z. Xiang, G. Kesidis, Adversarial learning in statistical classification: A comprehensive review of defenses against attacks, 2019, CoRR abs/1904.06292. arXiv:1904.06292. URL http://arxiv.org/abs/1904.06292.
[317] M. Ozdag, Adversarial attacks and defenses against deep neural networks: A survey, Procedia Comput. Sci. 140 (2018) 152–161, http://dx.doi.org/10.1016/j.procs.2018.10.315, Cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018. URL http://www.sciencedirect.com/science/article/pii/S1877050918319884.
[318] S. Thomas, N. Tabrizi, Adversarial Machine Learning: A Literature Review, 2018, pp. 324–334, http://dx.doi.org/10.1007/978-3-319-96136-1_26.
[319] S. Qiu, Q. Liu, S. Zhou, C. Wu, Review of artificial intelligence adversarial attack and defense technologies, Appl. Sci. 9 (5) (2019) http://dx.doi.org/10.3390/app9050909, URL http://www.mdpi.com/2076-3417/9/5/909.
[320] V. Duddu, A survey of adversarial machine learning in cyber warfare, 2018.
[321] W. Schwarting, J. Alonso-Mora, D. Rus, Planning and decision-making for autonomous vehicles, Annu. Rev. Control Robot. Auton. Syst. 1 (2018) http://dx.doi.org/10.1146/annurev-control-060117-105157.
[322] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, P.C. Cervera, A review of deep learning methods and applications for unmanned aerial vehicles, J. Sensors 2017 (2017) 3296874:1–3296874:13.
[323] A. Fridman, D.E. Brown, M. Glazer, W. Angell, S. Dodd, B. Jenik, J. Terwilliger, J. Kindelsberger, L. Ding, S. Seaman, H. Abraham, A. Mehler, A. Sipperley, A. Pettinato, B. Seppelt, L. Angell, B. Mehler, B. Reimer, Mit autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation, 2017, CoRR abs/1711.06976.
[324] S.D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y.H. Eng, D. Rus, M.H. Ang, Perception, planning, control, and coordination for autonomous vehicles, Machines 5 (1) (2017) http://dx.doi.org/10.3390/machines5010006, URL http://www.mdpi.com/2075-1702/5/1/6.
[325] G. von Zitzewitz, Survey of neural networks in autonomous driving, 2017.
[326] C. Badue, R. Guidolini, R.V. Carneiro, P. Azevedo, V.B. Cardoso, A. Forechi, L.F.R. Jesus, R.F. Berriel, T.M. Paixão, F.W. Mutz, T. Oliveira-Santos, A.F. de Souza, Self-driving cars: A survey, 2019, CoRR abs/1901.04407.
[327] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing [review article], IEEE Comput. Intell. Mag. 13 (2018) 55–75.
[328] D.W. Otter, J.R. Medina, J.K. Kalita, A survey of the usages of deep learning in natural language processing, 2018, CoRR abs/1807.10854.
[329] W. Khan, A. Daud, J. Nasir, T. Amjad, A survey on the state-of-the-art machine learning models in the context of nlp, 43 (2016) 95–113.
[330] S. Fahad, A. Yahya, Inflectional review of deep learning on natural language processing, 2018, http://dx.doi.org/10.1109/ICSCEE.2018.8538416.
[331] H. Li, Deep learning for natural language processing: advantages and challenges, Natl. Sci. Rev. 5 (1) (2017) 24–26, http://dx.doi.org/10.1093/nsr/nwx110, arXiv:http://oup.prod.sis.lan/nsr/article-pdf/5/1/24/24164446/nwx110.pdf. URL https://doi.org/10.1093/nsr/nwx110.
[332] Y. Xie, L. Le, Y. Zhou, V.V. Raghavan, Deep learning for natural language processing, 2018.
[333] S. Zhang, L. Yao, A. Sun, Deep learning based recommender system: A survey and new perspectives, ACM Comput. Surv. 52 (2019) 5:1–5:38.
[334] R. Mu, A survey of recommender systems based on deep learning, IEEE Access 6 (2018) 69009–69022, http://dx.doi.org/10.1109/ACCESS.2018.2880197.
[335] Z. Batmaz, A. Yurekli, A. Bilge, C. Kaleli, A review on deep learning for recommender systems: challenges and remedies, Artif. Intell. Rev. (2018) http://dx.doi.org/10.1007/s10462-018-9654-y.
[336] B.T. Betru, C.A. Onana, B. Batchakui, Deep learning methods on recommender system: A survey of state-of-the-art, 2017.
[337] R. Fakhfakh, B.a. Anis, C. Ben Amar, Deep learning-based recommendation: Current issues and challenges, Int. J. Adv. Comput. Sci. Appl. 8 (2017) http://dx.doi.org/10.14569/IJACSA.2017.081209.
[338] L. Zheng, A survey and critique of deep learning on recommender systems, 2016.

[339] O.Y. Al-Jarrah, P.D. Yoo, S. Muhaidat, G.K. Karagiannidis, K. Taha, Efficient machine learning for big data: A review, Big Data Res. 2 (3) (2015) 87–93, http://dx.doi.org/10.1016/j.bdr.2015.04.001, Big Data, Analytics, and High-Performance Computing. URL http://www.sciencedirect.com/science/article/pii/S2214579615000271.
[340] Y. Roh, G. Heo, S.E. Whang, A survey on data collection for machine learning: a big data - AI integration perspective, 2018, CoRR abs/1811.03402. arXiv:1811.03402. URL http://arxiv.org/abs/1811.03402.
[341] J. Qiu, Q. Wu, G. Ding, Y. Xu, S. Feng, A survey of machine learning for big data processing, EURASIP J. Adv. Signal Process. 2016 (1) (2016) 67, http://dx.doi.org/10.1186/s13634-016-0355-x.
[342] B. Jan, H. Farman, M. Khan, M. Imran, I.U. Islam, A. Ahmad, S. Ali, G. Jeon, Deep learning in big data analytics: A comparative study, Comput. Electr. Eng. 75 (2019) 275–287, http://dx.doi.org/10.1016/j.compeleceng.2017.12.009, URL http://www.sciencedirect.com/science/article/pii/S0045790617315835.
[343] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, J. Big Data 2 (1) (2015) 1, http://dx.doi.org/10.1186/s40537-014-0007-7.
[344] G. Marcus, Deep learning: A critical appraisal, 2018, CoRR abs/1801.00631.
[345] S. Sabour, N. Frosst, G.E. Hinton, Dynamic routing between capsules, in: NIPS 2017, 2017.
[346] G.E. Hinton, A. Krizhevsky, S.D. Wang, Transforming auto-encoders, in: ICANN, 2011.
[347] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, in: NIPS'16, Curran Associates Inc., USA, 2016, pp. 3637–3645, URL http://dl.acm.org/citation.cfm?id=3157382.3157504.
[348] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53, http://dx.doi.org/10.1145/963770.963772, URL http://doi.acm.org/10.1145/963770.963772.
[349] D.F. Polit, C.T. Beck, Resource manual for nursing research generating and assessing evidence for nursing practice/denise f. polit and beck,cheryl tatano, 2011.
[350] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, D. Koller, Max-margin classification of data with absent features, J. Mach. Learn. Res. 9 (2008) 1–21, URL http://dl.acm.org/citation.cfm?id=1390681.1390682.
[351] G. Chechik, G. Heitz, G. Elidan, P. Abbeel, D. Koller, Max-margin classification of incomplete data, in: NIPS, 2006.
[352] Y. Dong, C.-Y.J. Peng, Principled missing data methods for researchers, in: SpringerPlus, 2013.
[353] K. Mohan, G.V. den Broeck, A. Choi, J. Pearl, An efficient method for bayesian network parameter learning from incomplete data, in: ICML 2014, 2014.
[354] K. Mohan, J. Pearl, J. Tian, Graphical models for inference with missing data, in: NIPS, 2013.
[355] K. Hsu, S. Levine, C. Finn, Unsupervised learning via meta-learning, 2018, CoRR abs/1810.02334.
[356] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, 2017, CoRR abs/1703.03400 arXiv:1703.03400. URL http://arxiv.org/abs/1703.03400.
[357] A. Banino, C. Barry, B. Uria, C. Blundell, T.P. Lillicrap, P. Mirowski, A. Pritzel, M.J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N.C. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, D. Kumaran, Vector-based navigation using grid-like representations in artificial agents, Nature 557 (2018) 429–433.
[358] U.M. Erdem, M. Hasselmo, A goal-directed spatial navigation model using forward trajectory planning based on grid cells., Eur. J. Neurosci. 35 (6) (2012) 916–931.
[359] D. Bush, C. Barry, D. Manson, N. Burgess, Using grid cells for navigation, Neuron 87 (3) (2015) 507–520, http://dx.doi.org/10.1016/j.neuron.2015.07.006, URL http://www.sciencedirect.com/science/article/pii/S0896627315006285.
[360] I.R. Fiete, Y. Burak, T. Brookings, What grid cells convey about rat location, J. Neurosci. 28 (27) (2008) 6858–6871, http://dx.doi.org/10.1523/JNEUROSCI.5684-07.2008, arXiv:https://www.jneurosci.org/content/28/27/6858.full.pdf. URL https://www.jneurosci.org/content/28/27/6858.
[361] R.M. Cichy, D. Kaiser, Deep neural networks as scientific models, Trends Cogn. Sci. 23 (4) (2019) 305–317, http://dx.doi.org/10.1016/j.tics.2019.01.009, URL http://www.sciencedirect.com/science/article/pii/S1364661319300348.
[362] B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning, 2016, CoRR abs/1611.02167.
[363] C.J.C.H. Watkins, P. Dayan, Q-learning, Mach. Learn. 8 (3) (1992) 279–292, http://dx.doi.org/10.1007/BF00992698.
[364] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529 (2016) 484–503, URL http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.
[365] L. Kaiser, A.N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, J. Uszkoreit, One model to learn them all, 2017, ArXiv abs/1706.05137.
[366] M.L. Seltzer, J. Droppo, Multi-task learning in deep neural networks for improved phoneme recognition, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6965–6969.
[367] F.D. Atiah, M. Helbig, Effects of decision models on dynamic multi-objective optimization algorithms for financial markets, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 762–770.
[368] G. Montana, F. Parrella, Learning to trade with incremental support vector regression experts, in: E. Corchado, A. Abraham, W. Pedrycz (Eds.), Hybrid Artificial Intelligence Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 591–598.
[369] A.A. Rusu, N.C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, R. Hadsell, Progressive neural networks, 2016, ArXiv abs/1606.04671.