A Review of Deep Learning with Special Emphasis on Architectures, Applications and Recent Trends

Saptarshi Sengupta[a,1], Sanchita Basak[a], Pallabi Saikia[b], Sayak Paul[c], Vasilios Tsalavoutis[d], Frederick Atiah[e], Vadlamani Ravi[f,2], 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] *Datacamp, Inc.*
[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 500034, India*

Abstract

Deep learning has taken over - both in problems beyond the realm of traditional, hand-crafted machine learning paradigms as well as in capturing the imagination of the practitioner sitting on top of petabytes of data. While the public perception about the efficacy of deep neural architectures in complex pattern recognition tasks grows, sequentially up-to-date primers on the current state of affairs must follow. In this review, we seek to present a refresher of the many different stacked, connectionist networks that make up the deep learning architectures followed by automatic architecture optimization protocols using multi-agent approaches. Further, since guaranteeing system uptime is fast becoming an indispensable asset across multiple industrial modalities, we include an investigative section on testing neural networks for fault detection and subsequent mitigation. This is followed by an exploratory survey of several application areas where deep learning has emerged as a game-changing technology - be it anomalous behavior detection in financial applications or in financial time-series forecasting, predictive and prescriptive analytics, medical

⋆

[1]Corresponding author. Tel.: +1 615-6783419; . E-mail address: sengupta.sap@gmail.com

[2]Corresponding author. Tel.: +91 40 23294042; fax: +91 40 23535157. E-mail address: vravi@idrbt.ac.in

image analysis/processing or power systems research. The thrust of this review is on outlining emerging areas of application-oriented research within the deep learning community as well as to provide a handy reference to researchers seeking to embrace deep learning in their work for what it is: statistical pattern recognizers with unparalleled hierarchical structure learning capacity with the ability to scale with information.

Keywords: Neural Network Architectures, Supervised Learning, Unsupervised Learning, Testing Neural Networks, Applications of Deep Learning, Evolutionary Computation

---

1. Introuction

Artificial neural networks (ANNs), one of the most widely-used paradigms in computational intelligence, started out as an attempt to carry out synthetic mimicry of adaptive biological nervous systems in software and customized hardware implementations [1]. ANNs have made a strong resurgence as pattern recognition tools following pioneering work by a group of people [2] who demonstrated that stacked neural architectures can indeed learn complex, non-linear functional mappings given the right computational capabilities and that they scale with training data, unlike more traditional approaches. The intellectual neighbourhood has seen exponential growth, both in terms of academic and industrial research partly due the inherently trouble-free use of stacked neural architectures as blackbox implementations which eliminates the need to handcraft specifics of the problem but also due the state-of-the-art performances of the networks in applications which require deriving actionable insights from unstructured, high-dimensional data [3] [4] [5] [6] [7] [8]. This motivates this timely review which charts through the niche, starting with a brief description of artificial neural networks below:

1.1. ***What is an Artificial Neural Network?***

An artifical neural network is composed by many interconnected single units, or *'neurons'* and act as sequential or parallel information-processing-units. If one imagines a black-box created by stacking layers of these unitary neurons, the resulting architecture may carry out the following actions:

1. It may interact with the surrounding universe using some of its atomic units to receive information (these units are known to be part of the *input layers* of the *neural network*).

2. It may pass information back-and-forth among the stacked layers within the black-box and process the information by invoking certain *design goals* and *learning rules* (these units are known to be parts of the *hidden layers* of the *neural network*).

3. It may relay information out to the surrounding universe using some of its atomic units (these units are known to be part of the *output layers* of the *neural network*).

Each neuron is activated if the incoming signal is larger than some *threshold* and propagates a signal to all neurons connected to it. The connection mechanism acts like a filter - it weighs the signal with either a positive or negative weight, drawing parallels from the excitation and inhibition processes in biological neural systems. In general, the system response of the black-box to an excitation from the surrounding universe depends on the details of the connectivity of internal units and the distribution of weights.

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

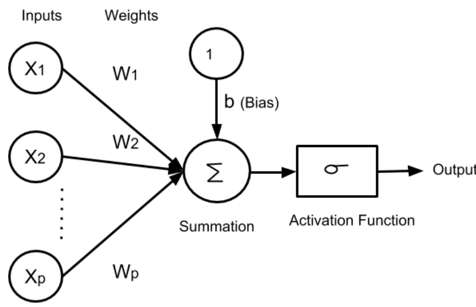

Figure 1: The Perceptron Learning Model

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 [9] 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 [10] [11] [12]. 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.

Table 1: Some Key Advances in Neural Networks Research

People Involved	**Contribution**
McCulloch & Pitts	ANN models with adjustible weights (1943) [13]
Rosenblatt	The Perceptron Learning Algorithm (1957) [14]
Widrow and Hoff	Adaline (1960), Madaline Rule I (1961) & Madaline Rule II (1988)[15] [16]
Minsky & Papert	The XOR Problem (1969) [17]
Werbos (Doctoral Dissertation)	Backpropagation (1974) [18]
Hopfield	Hopfield Networks (1982) [19]
Rumelhart, Hinton & Williams	Renewed interest in backpropagation: multilayer adaptive backpropagation (1986) [20]
Vapnik, Cortes	Support Vector Networks (1995) [21]
Hochreiter & Schmidhuber	Long Short Term Memory Networks (1997) [22]
LeCunn et. al.	Convolutional Neural Networks (1998) [23]
Hinton & Ruslan	Hierarchical Feature Learning in Deep Neural Networks (2006) [24]

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 [25], Tensorflow [26], Caffe [27] , Chainer [28], Keras [29], BigDL [30] 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 momemtum or Nesterov momentum), Particle Swarm Optimization, Differential Evolution, etc.

Table 2: A Collection of Popular Deployment Platforms

Software Platform	**Purpose**
Tensorflow [26]	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 [31]	GPU compatible Python library with tight integration to NumPy involves smooth mathematical operations on multidimensional arrays. url: `http://deeplearning.net/software/theano/`
CNTK [32]	Microsoft Cognitive Toolkit (CNTK) is a Deep Learning Framework describing computations through directed graphs. url: `https://www.microsoft.com/en-us/cognitive-toolkit/`
Keras [29]	It runs on top of Tensorflow, CNTK or Theano compatible to be deployed in CPU and GPU. url: `https://keras.io/`
PyTorch [25]	Distributed training and performance evaluation platform integrated with Python supported by major cloud platforms. url: `https://pytorch.org/`
Caffe [27]	Convolutional Architecture for Fast Feature Embedding (Caffe) is a Deep Learning framework with focus on image classsification and segmentation and deployable in both CPU and GPU. url: `http://caffe.berkeleyvision.org/`
Chainer [28]	Supports CUDA computation and multiple GPU implementation. url: `https://chainer.org/`
BigDL [30]	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`

1.4. *Contributions made by this article*

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 ot 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:

  · Anomaly Detection in Financial Services,
  · Financial Time Series Forecasting,
  · Prognostics and Health Monitoring,
  · Medical Imaging and
  · Power Systems

The rest of the paper is organized 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 dissection of the general successes and pitfalls of the field as of now and concludes the article.

2. Deep architectures: Working mechanisms

There are numerous deep architectures available in the literature. The Comparison of architectures is difficult as different architectures have different advantages based on the application and the characteristics of the data involved, for example, In vision, Convolutional Neural Networks [23], for sequences and time series modelling Recurrent neural networks [33] is prefered. However, deep learning is a fast evolving field. In every year various architectures with various learning algorithms are developed to endure the need to develop human-like efficient machines in different domains of application.

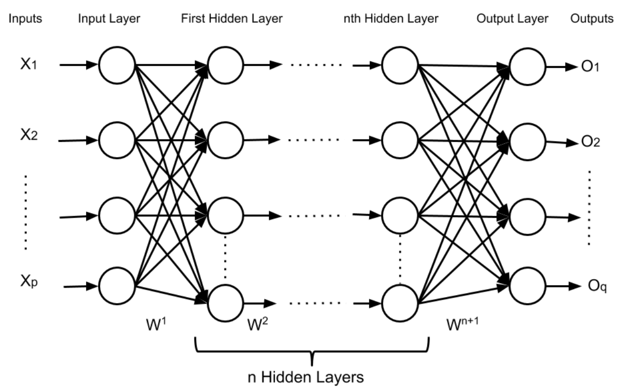

Figure 2: Deep Feed-forward Neural Network with n Hidden layers, p input units and q output units with weights W.

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 [34]. An example of Deep Feed-Forward Network with n hidden layers is provided in Figure 2. Multiple hidden layers help in modelling complex nonlinear relation more efficiently compared to the shallow architecture. A complex function can be modelled 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 [35]. 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 [36] is the most common learning algorithm used to train this model. The algorithm first initialises the weights randomly, and then the weights are tuned to minimise 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 nonlinearity and ultimately compare the computed output with the actual output of the corresponding input. In the backward pass, the error derivatives with respect to the network parameters are back propagated to adjust the weights in order to minimise 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,

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

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

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

Many issues like overfitting, trapped in local minima and vanishing gradient issues can arise if a deep neural network is trained naively. This was the reason; neural network was forsaken by the machine learning community in the late 1990s. However, in 2006 [24, 37], with the advent of unsupervised pre-training approach in deep neural network, the neural network is revived again to be used for the complex tasks like vision and speech. Lately, many other techniques like l1, l2 regularisation [38], dropout [39], batch normalisation [40], good set of weight initialisation techniques [41, 42, 43, 44] and good set of activation functions [45] are introduced to combat the issues in training deep neural networks.

2.2. Restricted Boltzmann Machines

Restricted Boltzmann Machine (RBM) [46] can be interpreted as a stochastic neural network. It is one of the popular deep learning frameworks 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 [47]. However, it gets popularised by Hinton in 2002 [48] with the advent of the improved training algorithm to RBM. After that, it got a wide application in various tasks like representation learning [49], dimensionality reduction [50], prediction problems [51]. However, deep belief network training using the RBM as building block [24] was the most prominent application in the history of RBM that provides the starting of deep learning era. Recently RBM is getting immense popularity in the field of collaborative filtering [52] due to the state of the art performance in Netflix.

Restricted Boltzmann Machine is a variation of Boltzmann machine with the restriction in the intra-layer connection between the units, and hence called restricted. It is an undirected graphical model containing two layers, visible and hidden layer, forms a bipartite graph. Different variations of RBMs have been introduced in literature in terms of improving the learning algorithms, provided the task. Temporal RBM [53] and conditional RBM [54] proposed and applied to model multivariate time series data and to generate motion captures, Gated RBM [55] to learn transformation between two input images, Convolutional RBM [56, 57] to understand the time structure of the input time series, mean-covariance RBM [58, 59, 60] to represent the covariance structure of the data, and many more like Recurrent TRBM [61], factored conditional RBM (fcRBM) [62]. Different types of nodes like Bernoulli, Gaussian [63] are introduced to cope with the characteristics of the data used. However, the basic RBM modelling concept 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 Figure 3.

The joint probability distribution of an standard RBM can be defined with

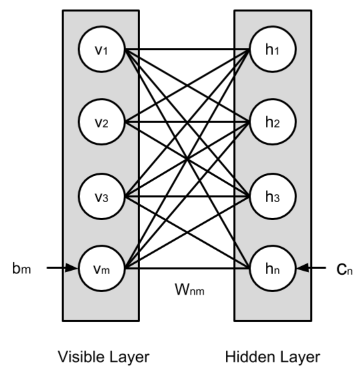

Figure 3: RBM with m visible units and n hidden units

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 normalisation 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 connection make the RBM hidden layer variables independent given the states of the visible layer variables and vice versa. This easy down the complexity of modelling the probability distribution and hence the probability distribution of each variable can be represented by conditional probability distribution as given below:

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

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

RBM is trained to maximise the expected probability of the training samples. Contrastive divergence algorithm proposed by Hinton [48] is popular for the training of RBM. The training brings the model to a stable state by minimising its energy by updating the parameters of the model. The parameters can be updated using the following equations:

$$\Delta w_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{model}) \quad (7)$$

$$\Delta b_i = \epsilon(< v_i >_{data} - < v_i >_{model}) \quad (8)$$

$$\Delta c_j = \epsilon(< h_j >_{data} - < h_j >_{model}) \quad (9)$$

Where, ϵ is the learning rate, $< . >$ data , $< . >$ 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, 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 [64] with three hidden layers is shown in Figure 4.

Inference in DBN is an intractable problem due to the explaining away effect in the latent variable model. However, in 2006 Hinton [24] 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

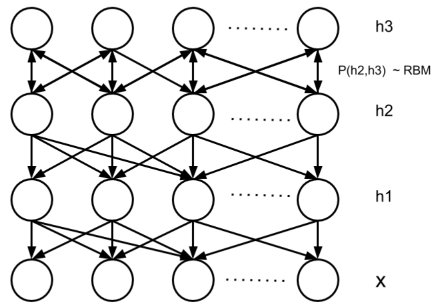

Figure 4: DBN with input vector **x** with 3 hidden layers

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 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:

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

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 [65]. The DBN training proposed in 2006 [24] by Hinton led to the deep learning era of today and revived the neural network. This was the first deep architecture in the history able to train efficiently. Before that, it was almost impossible to train deep architectures. Deep architectures build by initialising the weights with DBN, outperformed the kernel machines, that was in the research landscape at that time. DBN, along with its use as generative models, significantly applied as discrimination model by appending a discrimination layer at the end and fine-tuning the model using the target labels provided [2]. In most of the applications, this approach of pretraining a deep architecture led to the state of the performance in discriminative model [66, 24, 37, 67, 50] like in recognising handwritten digits, detecting pedestrians, time series prediction etc. even when the number of labelled data was limited [68]. It has got immense popularity in acoustic modelling [69] recetly as the model could provide upto 20% improvement over state of the art models, Hidden Markov Model, Gaussian Mixture Model. The approach creates feature detectors hierarchically as features of features in pretraining that provide a good set of initialised weights to the discriminative model. The initialised weights are in a region near the optimal weights that can improve both modelling and the convergence in fine-tuning [66, 70]. DBN has been used as an initialised model in classification in many applications like in phone recognition [58], computer vision [59] where it is used for the training of higher order factorized Boltzmann machine, speech recognition [71, 72, 73] for pretraining DNN, for pretraining of deep convolutional neural network (CNN) [56, 74, 57]. The improved performance is due to the ability to learn some abstract features by 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 [60, 75, 76].

2.4. ***Autoencoders***

Autoencoder is a three-layer neural network, as shown in Figure 5, that tries to reconstruct its input in 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, tries to encode or represent the input in a more compact form. It shares the same idea as RBM, but it typically uses deterministic distribution instead of stochastic units with particular distribution as in the case of RBM.

Like feedforward neural network, autoencoder is typically trained using back-propagation 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 transposed of each other. The encoding and

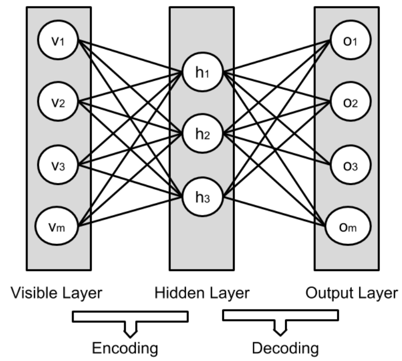

Figure 5: Autoencoder with 3 neurons in hidden layer

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:

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

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

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.

Autoencoder with multiple hidden layers forms a deep autoencoder. Similar like in deep neural network, autoencoder training may be difficult due to multiple layers. This can be overcome by training each layer of deep autoencoder as a simple autoencoder [24, 37]. The approach has been successfully applied to encode documents for faster subsequent retrieval [77], image retrieval, efficient speech features [78] etc. As like RBM stacking to form DBN [24] for layerwise pretraining of DNN, autoencoder [37] along with sparse encoding energy-based model [67] are independently developed at that time. They both 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 like in image recognition and dimensionality reduction in MNIST [50, 78, 79], multimodal learning in speech and video images [80, 81] and many more. Autoencoder has got immense popularity as generative model in recent years [34, 82]. Non Probabilistic and non-generative nature of conventional autoencoder has been generalised to generative modelling [83, 38, 84, 85, 86] that can be used to generate the samples from the network meaningfully.

Several variations of autoencoders are introduced with quite different properties and implementation to learn more efficient representation of data. One of the popular variation of autoencoder that is robust to input variations is denoising autoencoder [85, 38, 86]. The model can be used for good compact representation of input with the number of hidden layers less than the input layer. It can also be used to perform robust modelling of the input distribution with higher number of neurons in the hidden layer. The robustness in denoising autoencoder is achieved by introducing dropout trick or by introducing some gaussian noise to the input data [87, 88] or to the hidden layers [89]. The approach helps in many many ways to improve performance. It virtually increasing the training set hence reduce overfitting, and make robust representation of the input. Sparse autoencoder [89] is introduced in a consideration to allow larger number of hidden units than the visible units to make it easier and efficient representation of the input distribution in the hidden layer. The larger hidden layer represent the input representation by turning on and off the units in the hidden layer. Variational autoencoder [82, 90] that uses quite the similar concept as RBM, learn stochastic distribution of latent variables instead

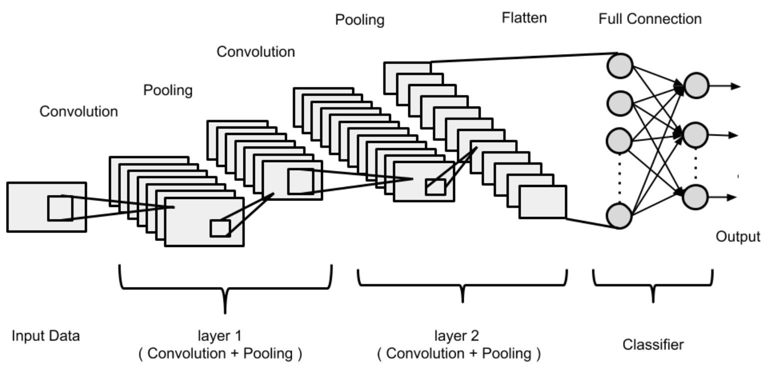

Figure 6: Convolution and Pooling Layers in a CNN

of deterministic distribution. Transforming autoencoders [91] proposed as a autoencoder with transformation invariant property. The encoded features of the autoencoder can effectively reflect the transformation invariant property. The encoder is applied in image recognition [91, 92] purpose that contains capsule as the building block. Capsule is an independent sub-network that extracts local features within a limited window of viewing to understand if a feature entity is present with certain probability. Pretraining for CNN using regularised deep autoencoder is very much popularised in recent years in computer vision works. Robust models of CNN is obtained with denoising autoencoder [84] and sparse autoencoder with pooling and local contrast normalization [93] which provides not only translation-invariant features but also scaling and out-of-plane rotation invariant features.

2.5. ***Convolutional Neural Networks***

Convolutional Neural Networks are a class of neural networks that are extremely good for processing images. Although its idea was proposed way back in 1998 by LeCun et. al in their paper entitled ”Gradient-based learning applied to document recognition” [94] but the deep learning world actually saw it in action when Krizhevsky et. al were able win the ILSVRC-2012 competition. The

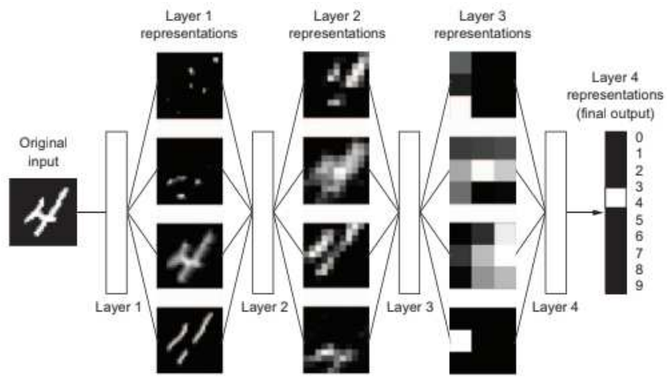

Figure 7: Representations of an image of handwritten digit learned by CNN

architecture that Krizhevsky et. al proposed is popularly known as AlexNet [95]. This remarkable win started the new era of artificial intelligence and the computation community witnessed the real power of CNNs. Soon after this, several architectures have been proposed and still are being proposed. And in many cases, these CNN architectures have been able to beat human recognition power as well. It is worth to note that, The deep learning revolution actually with the usage of Convolutional Neural Networks (CNNs). CNNs are are extremely useful for a set computer vision related tasks such as image detection, image segmentation, image classification and so on and all of these tasks are practically well aligned. On a very high level, deep learning is all about learning data representations and in order to do so deep learning systems typically breaks down complex representations into a set of simpler representations. As mentioned earlier, CNNs are particularly useful when it comes to images as images have a special spatial property in their formations. An image has several characteristics like edges, contours, strokes, textures, gradients, orientation, colour. A CNN breaks down an image in terms of simple properties like these and learn them as representations in different layers [96]. Figure 7 is a good representative of this learning scheme.

The layers involved in any CNN model are the convolution layers and the

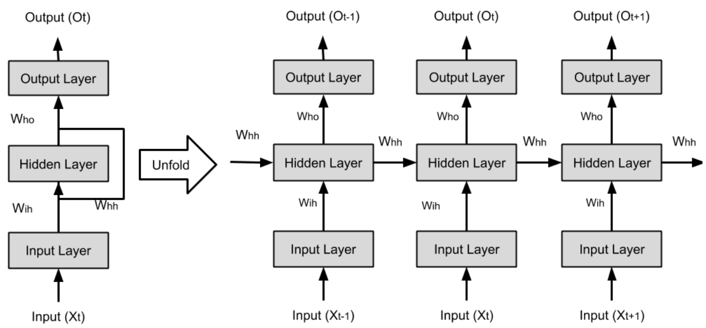

Figure 8: A Recurrent Neural Network Architecture

subsampling/pooling layers which allow the network learn filters that are specific to specific parts in an image. The convolution layers help the network retain the spatial arrangement of pixels that is present in any image whereas the pooling layers allow the network to summarize the pixel information [97]. There are several CNN architectures ZFNet, AlexNet, VGG, YOLO, SqueezeNet, ResNet and so on and some these have been discussed in section 2.8.

2.6. ***Recurrent Neural Networks***

Although Hidden Markov Models (HMM) can express time dependencies, they become computationally unfeasible in the process of modelling long term dependencies which RNNs are capable of. A detailed derivation of Recurrent Neural Network from differential equations can be found in [98]. 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. During the backpropagation of errors across multiple 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 [99]. 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.

LSTM networks with peephole connections [100] 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) [101] merging the hidden and the cell state. In [102] Sak et al., came up with training LSTM RNNs in a distributed way on multicore CPU using asynchronus SGD (Stochastic Gradient Descent) optimization for the purpose of acoustic modelling. 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., [103] 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., [104] 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.

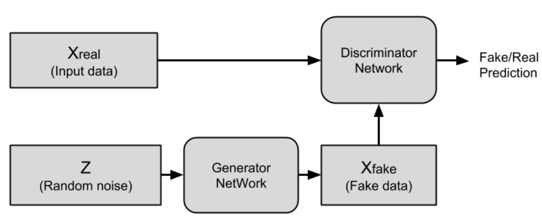

Figure 9: A Generative Adversarial Network Architecture

2.7. ***Generative Adversarial Networks***

Goodfellow et al., [105] 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.

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% probalilty 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.

In [106], 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 [107], 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.

Wu et al., [108] 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., [109] 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 [110]. 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. ***Recent Deep Architectures***

When it comes to deep learning and computer vision, datasets like Cats and Dogs, ImageNet, CIFAR-10, MNIST are used for benchmarking purposes. Throughout this section, the ImageNet dataset is used for the purpose of benchmarking results as it is more generalized than the other datasets just mentioned. Every year a competition named ILSVRC (ImageNet Large Scale Visual Recognition Competition) is organized (which is an image classification competition) which based on the ImageNet dataset and it is widely accepted by the deep learning community [111].

Several deep neural network architectures have been proposed in the literature and still are being proposed with an objective of achieving general artificial intelligence. LeNet architecture, for example was proposed by Lecun et. al in 1998s and it was originally proposed as a digit classification model. Later, LeNet has been incorporated to identify handwritten numbers on cheques [94]. Several architectures have been proposed after LeNet among which AlexNet certainly deserves to be the most notable mentions. It was proposed by Krizhevsky et. al in 2012 and AlexNet was able to beat all the competitors of the ILSVRC challenge. The discovery of AlexNet marks a significant turn in the history of deep learning for several reasons such as AlexNet incorporated the dropout regularization which was just developed by that time, AlexNet made use of efficient GPU computing for reducing the training time which was first of its kind back in 2012 [95]. Soon after AlexNet , ZFNet was proposed by Zeiler et. al in the year of 2013 and showed state-of-the-art results on the ILSVRC challenge. It was an enhancement of the AlexNet architecture. It uses expanded mid convolution layers and incorporates smaller strides and filters in the first convolution layer for capturing the pixel information in a great detail [112]. In 2014, Google researchers came with a better model which is known as GoogleNet or the Inception Network and won the ILSVRC 2014 challenge. The main catch of this architecture is the inception layer which allows convolving in parallel with different kernel sizes. This is turn allows to learn the smaller pixel information of an image in a better way [113]. It's worth to mention the VGGNet (also called VGG) architecture here. It was the runners' up in the ILSVRC 2014 challenge and was proposed by Simonyan et. al. VGG uses a 3X3 kernel throughout its entire architecture and ahieves tremendous generalization with this fixation [114]. The inner of the ILSVRC 2015 challenge was the ResNet architecture and was proposed by He et. al. This architecture is more formally known as Residual Networks and is deeper than the VGG architecture while still being less complex in the VGG architecture. ResNet was able to beat human performance on the ImageNet dataset and it is still being quite actively used in production [115] [116].

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 [117], 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 CIFER-10, CIFER-100, MNIST and Fashion MNIST dataset. On the other hand, Garro et al., [118] 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 well known Back-Propagation and Levenberg-Marquardt algorithms. Das et al. [119], 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. [120], used Variable-length Particle Swarm Optimization for automatic evolution of deep Convolutional Neural Network Architectures for image classification purposes. They proposed novel encoding strategy 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. Thus several variants of PSO along with its hybridised versions [121] as well as a host of recent swarm intelligence algorithms such as Quantum Double Delta Swarm Algorithm (QDDS) [122] and its chaotic implementation [123] proposed by Sengupta et al. can be used, among others for automatic generation of architectures used in Deep Learning applications.

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 [124] using an endtoend learned mean feature embedding as state information. The research concluded that an endtoend 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 [125] and in [126] using Principle component analysis. In [127] 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, [128] 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 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. [129], 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, [130], 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., [131] deals with generating scenarios leading to unexpected behaviors by introducing perturbations in the testing conditions. For identifying fasification 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 [132] 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 [133]:

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

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. [134] 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 [132] 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 multifaceted 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.

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 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

Table 3: Distribution of Articles by Application Areas

Application Area	**Authors**
Fraud Detection in Financial Services	Pumsirirat et al. [135], Schreyer et al. [136], Wang et al. [137], Zheng et al. [138], Dong et al. [139], Gomez et al. [140], Rymantubb et al. [141], Fiore et al. [142]
Financial Time Series Forecasting	Cavalcante et al. [143], Li et al. [144], Fama et al. [145], Lu et al. [146], Tk & Verner [147], Pandey et al. [148], Lasfer et al. [149], Gudelek et al. [150], Fischer & Krauss [151], Santos Pinheiro & Dras [152], Bao et al. [153], Hossain et al. [154], Calvez and Cliff [155]
Prognostics and Health Monitoring	Basak et al. [156], Tamilselvan & Wang [157], Kuremoto et al. [158], Qiu et al. [159], Gugulothu et al. [160], Filonov et al. [161], Botezatu et al. [162]
Medical Image Processing	Suk, Lee & Shen [163], van Tulder & de Bruijne [164], Brosch & Tam [165], Esteva et al. [166], Rajaraman et. al. [167], Kang et al. [168], Hwang & Kim [169], Andermatt et al. [170], Cheng et al. [171], Miao et al. [172], Oktay et al. [173], Golkov et al. [174]
Power Systems	Vankayala & Rao [175], Chow et al. [176], Guo et al. [177], Bourguet & Antsaklis [178], Bunn & Farmer [179], Hippert et al. [180], Kuster et al. [181], Aggarwal & Song [182], Zhai [183], Park et al. [184], Mocanu et al. [185], Chen et al. [186], Bouktif et al. [187], Dedinec et al. [188], Rahman et al. [189], Kong et al. [190], Dong et al. [191], Kalogirou et al. [192], Wang et al. [193], Das et al. [194], Dabra et al. [195], Liu et al. [196], Jang et al. [197], Gensler et al. [198], Abdel-Nasser et al. [199], Manwell et al. [200], Marugán et al. [201], Wu et al. [202], Wang et al. [203], Wang et al. [204], Feng et al. [205], Qureshi et al. [206]

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)[135] 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) [136] 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) [137] 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)[138] 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)[139] 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)[140] 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 practitioners 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) [141] 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) [142] 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.

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 [143]. On the other hand, NN and deep learning models have shown great success in forecasting financial time series [144] despite the contradictory report by efficient market hypothesis (EMH) [145], 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 [146].

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 [143]. 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 [147]. According to the experimental analysis done by Pandey el at [148], showed that MFF with Bayesian learning performed better than MFF learning with back propagation for the FOREX market.

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 el at [149] 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 [149], 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 [150].

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 el at [150] used 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 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 [151] 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 [145] 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 [143]. Santos et al [152] 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 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 [153] combined wavelet transform, stacked autoencoders and LSTM for stock price prediction. 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 [154], 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 [155] did introduce a new approach on how to trade on the stock market with DLNN model. DLNN model learn or observe the trading behaviors of traders. The author used a limit-order-book (LOB) and quotes made by successful traders (both automated and humans) as input data. DLNN was able to learn and outperformed both human traders and automated traders. This approach of learning might be the 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.

The recent works on device health forecasting are as follows: Basak et al. [156] carried on Remaining Useful Life (RUL) prediction of hard disks along with discussions on effective feature normalization strategies on Backblaze hard disk data. Deep Belief Networks (DBN) based multisensor health diagnosis methodology has been proposed in [157] and employed in aircraft engine and electric power transformer health diagnosis to show the effectiveness of the approach.

Kuremoto et al., [158] applied DBN composed of two Restricted Botzmann Machines (RBM) to capture the input feature distribution and then optimized the size of the network and learning rate through Particle Swarm Optimization for forecasting purposes with time series data. Qiu et al., [159] 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. Gugulothu et al. [160] 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 idea of using RNNs to capture intricate dependencies among various time cycles of sensor observations is emphasized in [161] for prognostic applications. Botezatu et al., came up with some rules for directly identifying the healthy or unhealthy state of a device in [162], employing a disk replacement prediction algorithm with changepoint detection applied to time series Backblaze data. Thus deep learning architectures have been extensively used in prognostics starting to replace some of the model driven 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. [207] 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. [208]. A concise overview of recent work in some of these canonical tasks follows.

The purpose of image/exam classification jobs is to identify the presence of a disease based on the images of medical examinations. Over the last few years, various neural network architectures have been used in this field including stacked auto-encoders applied to diagnosis of Alzheimers disease and mild cognitive impairment, exploiting the latent non-linear complicated relations among various features [163], Restricted Boltzmann Machines applied to Lung CT analysis combining generative as well as discriminative learning techniques [164], Deep Belief Networks trained on three dimensional medical images [165] etc. Recently, the the trend of using Convolutional Neural Networks in the field of image processing has been observed. In 2017, Esteva et al. [166] used and fine-tuned the Inception v3 [209] model to classify clinical images pertaining to skin cancer examinations into benign and malignant variants. Validated of experiments was carried out by testing model performance against a good number of dermatologists. In 2018, Rajaraman et. al [167] used specialized CNN architectures like ResNet for detecting malarial parasites in thin blood smear images. Kang et al. [168] 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.

Object/lesion detection aims to identify different parts/lesions in an image.

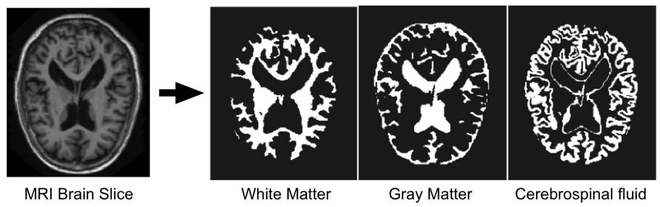

Figure 10: MRI Brain Slice and its different segmentation [211]

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 [210]. 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 [169].

Segmentation happens to be one of the most common subjects of interest when it comes to application of Deep Learning in the domain of 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. A demonstration is shown in Figure 10, where different segmented parts of an MRI Brain Slice along with the original slice are considered. Segmentation includes both the local and global context of pixels with respect to a given image and the performance of a segmentation model can suffer from inconsistencies due to class imbalances. This makes the task of segmentation a difficult one. The most widely-used CNN architecture for medical image segmentation is U-Net which was proposed by Ronneberger et al. [212] 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. [170] 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. [171] used multi-modal stacked denoising autoencoder to compute effective similarity measure among images using normalized mutual information and local cross correlation. On the other hand, Miao et al. [172] developed CNN regressors to directly evaluate the registration transformation parameters. In addition to these, image generation and enhancement techniques have been discussed in [173], [174].

5.5. ***Power Systems***

Artificial Neural Networks (ANN) have rapidly gained popularity among power system researchers [175]. Since their introduction to the power systems area in 1988 [176], 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 datasets and often outperform traditional machine learning methods in problems related to the power sector [177]. 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 [178]. A small decrease in load forecasting error may result in significant reduction of the total operation cost of the power system [179]. Among the Artificial Intelligence techniques applied for load forecasting, methods based on ANN have undoubtedly received the largest share of attention [180]. A basic reason for their popularity lies on the fact that ANN techniques are well-suited for energy forecast [181]; they may obtain adequate estimations in cases where data is incomplete [182] and can consistently deal with complex non-linear problems [183]. Park et al. [184], 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 [185]. Thus, DL methods are gradually gain increased popularity due to their ability to capture data behaviour when considering complex non-linear patterns and large amounts of data. In [186], an end-to-end model based on deep residual neural networks is proposed for hourly load forecasting of a single day. 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 a deep neural network structure constructed by residual blocks. The efficiency of the proposed model was demonstrated on data-sets from the North-American Utility and ISO-NE. In [187], 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 efficient performance of the proposed structure was verified using electricity consumption data of the France Metropolitan. Mocanu et al. [185] utilized two deep learning approaches based on Restricted Boltzman Machines (RBM), i.e. conditional RBM and factored conditional RBM, for single-meter residential load forecasting. The method was benchmarked against several shallow ANN architectures and a Support Vector Machine approach, demonstrating increased efficiency compared to the competing methods. Dedinec et al. [188] 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. Rahman et al. [189] 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 [190]. The researchers have showed that their method outperforms other state-of-the-art methods for load forecasting. In [191] 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 modelling 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 [192]. 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 [193]. Among the various renewable energy sources, wind and solar energy have gained more popularity due to their potential and high availability [194]. As a result, in recent years the research endeavours 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 [195]. However, due to the chaotic and erratic nature of the weather systems, the power output of PV energy systems is intermittent, volatile and random [196]. 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 [197]. For these reasons, the accuracy of forecasting of PV power output plays a major role in ensuring optimum planning and modelling of PV plants. In [193] 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. 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 data-sets obtained from two PV farms in Belgium, exhibiting high forecasting stability and robustness. In Gensler et al. [198], 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 [199] 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 [200]. ANN have been widely employed for processing large amounts of data obtained from data acquisition systems of wind turbines [201]. In recent years, many approaches based on DL architectures have been proposed for the prediction of the power output of wind power systems. In [202], 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. [203] proposed a wind power forecasting method based on Wavelet Transform, CNN and ensemble technique. Their method was compared with the persistence method and two shallow ANN architectures, i.e. Back-Propagation ANN (BPANN) and Support Vector Machine, on data sets collected from wind farms in China. The results validate that their method outperforms the benchmark approaches in terms of reliability, sharpness and overall skill. In [204] 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. A data-driven multi-model wind forecasting methodology with deep feature selection is proposed in [205]. In particular, a two layer ensemble technique is developed; the first layer comprises multiple machine learning models, which generate individual forecasts. In the second layer a blended algorithm is utilized to merge the forecasts derived during the first stage. Numerical results validate the efficiency of the proposed methodology compared to models employing a single algorithm. Finally, in [206] an approach is proposed for wind power forecasting, which combines deep Autoencoders, DBN and the concept of transfer learning. 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, and derives better results in terms of MAE, RMSE and SDE compared to the benchmark algorithms.

6. Discussions

In this paper we presented several Deep Learning architectures starting from the foundational architectures up to the recent developments covering the aspect of their modifications and evolution over time as well as applications to specific domains. We discussed the blend of swarm intelligence in Deep Learning approaches and how the influence of one enriches other when applied to real world problems. The vastly growing use of deep learning architectures specially in safety critical systems brings us to the question, how reliable the architectures are in providing decisions even in presence of adversarial scenarios. To address this, we started by giving an overview of testing neural network architectures, various methods for adversarial test generation as well as countermeasures to be adopted against adversarial examples. Next we moved on to specific applications of deep learning including Medical Imaging, Prognostics and Health Management, Applications in Financial Services, Financial Time Series Forecasting and lastly the applications in Power Systems.

In conclusion, we highlight a few open areas of research and elaborate on some of the existing lines of thoughts and studies in addressing challenges that lie within.

- **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 [213] if not trained with billions of training samples. Also the large reliance on Convolutional Neural Networks(CNNs) especially for video recognition purposes could face exponential ineffeciency leading to their demise [214] which can be avoided by capsules [215] 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 [216] is also bringing 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 is a current research thrust.

- **Adopting unsupervised approaches:** A major thrust is towards combining deep learning with unsupervised learning methods. Systems developed to set their own goals [213] and develop problem-solving approaches in its way towards exploring the environment are the future research directions surpassing supervised approaches requiring lots of data apriori. So, the thrust of AI research including Deep Learning is towards Meta Learning, i.e., learning to learn which involves automated model designing and decision making capabilities of the algorithms. It optimizes the ability to learn various tasks from fewer training data[217].

- **Influence of cognitive meuroscience:** Inspiration drawn from cognitive neuroscience, developmental psychology to decipher human behavioral pattern are able to bring major breakthrough in applications such as enabling artificial agents learn about spatial navigation on their own which comes naturally to most living beings [218].

- **Neural networks and reinforcement learning:** Meta-modeling approaches using Reinforcement Learning(RL) are being used for designing problem specific Neural Network architectures. In [219] the authors introduced MetaQNN, a RL based meta-modeling algorithm to automatically generate CNN architectures for image classification by using Q-learning

[220] with ϵ greedy exploration. AlphaGo, 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.

This review has aimed at aiding the beginner as well as the practitioner in the field make informed choices and has made an in-depth analysis of some recent deep learning architectures as well as an exploratory dissection of some pertinent application areas. It is the authors' hope that readers find the material engaging and informative and openly encourage feedback to make the organization and content of this article more aligned along the lines of a formal extension of the literature within the deep learning community.

References

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

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

[3] S. Lawrence, C. L. Giles, , A. D. Back, Face recognition: a convolutional neural-network approach, IEEE Transactions on Neural Networks 8 (1) (1997) 98–113. doi:10.1109/72.554195.

[4] 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. doi:10.1109/CVPR.2015.7298965.

[5] 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. doi:10.1109/TPAMI.2016.2599174.
URL https://doi.org/10.1109/TPAMI.2016.2599174

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

[7] 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. doi:10.1109/CVPR.2017.545.

[8] 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 Transactions on Pattern Analysis and Machine Intelligence 39 (7) (2017) 1320–1334. `doi:10.1109/TPAMI.2016.2587642`.

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

[10] G. Marcus, Deep Learning: A Critical Appraisal, arXiv e-prints (2018) arXiv:1801.00631`arXiv:1801.00631`.

[11] 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. `doi:10.1109/EuroSP.2016.36`.

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

[13] W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, The bulletin of mathematical biophysics 5 (4) (1943) 115–133. `doi:10.1007/BF02478259`.
URL `https://doi.org/10.1007/BF02478259`

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

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

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

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

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

[19] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79 (8) (1982) 2554–2558. arXiv:https://www.pnas.org/content/79/8/2554.full.pdf, doi:10.1073/pnas.79.8.2554.
URL https://www.pnas.org/content/79/8/2554

[20] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, MIT Press, Cambridge, MA, USA, 1986, Ch. Learning Internal Representations by Error Propagation, pp. 318–362.
URL http://dl.acm.org/citation.cfm?id=104279.104293

[21] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297. doi:10.1023/A:1022627411411.
URL https://doi.org/10.1023/A:1022627411411

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

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

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

[25] 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.

[26] 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, software available from tensorflow.org (2015).
URL `http://tensorflow.org/`

[27] 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, MM '14, ACM, New York, NY, USA, 2014, pp. 675–678. `doi:10.1145/2647868.2654889`.
URL `http://doi.acm.org/10.1145/2647868.2654889`

[28] 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`

[29] F. Chollet, et al., Keras, `https://github.com/fchollet/keras` (2015).

[30] 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, CoRR abs/1804.05839.

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

[32] 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, KDD '16, ACM, New York, NY, USA, 2016, pp. 2135–2135. `doi:10.1145/2939672.2945397`. URL `http://doi.acm.org/10.1145/2939672.2945397`

[33] 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.

[34] L. Deng, D. Yu, et al., Deep learning: methods and applications, Foundations and Trends® in Signal Processing 7 (3–4) (2014) 197–387.

[35] Y. Bengio, et al., Learning deep architectures for ai, Foundations and trends® in Machine Learning 2 (1) (2009) 1–127.

[36] 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).

[37] 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.

[38] 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.

[39] 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.

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

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

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

[43] 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.

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

[45] A. L. Maas, A. Y. Hannun, A. Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. icml, Vol. 30, 2013, p. 3.

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

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

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

[49] 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.

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

[51] 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.

[52] 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.

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

[54] 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.

[55] 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.

[56] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 609–616.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] 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.

[62] 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.

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

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

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

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

[67] 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.

[68] 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.

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

[70] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, Journal of Machine Learning Research 11 (Feb) (2010) 625–660.

[71] S. M. Siniscalchi, J. Li, C.-H. Lee, Hermitian polynomial for speaker adaptation of connectionist speech recognition systems, IEEE Transactions on Audio, Speech, and Language Processing 21 (10) (2013) 2152–2161.

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

[73] 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.

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

[75] J. Susskind, V. Mnih, G. Hinton, et al., On deep generative models with applications to recognition, in: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 2857–2864.

[76] V. Stoyanov, A. Ropson, J. Eisner, Empirical risk minimization of graphical model parameters given approximate inference, decoding, and model structure, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 725–733.

[77] R. Salakhutdinov, G. Hinton, Semantic hashing, International Journal of Approximate Reasoning 50 (7) (2009) 969–978.

[78] 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.

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

[80] 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.

[81] 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.

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

[83] G. Alain, Y. Bengio, What regularized auto-encoders learn from the data-generating distribution, The Journal of Machine Learning Research 15 (1) (2014) 3563–3593.

[84] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE transactions on pattern analysis and machine intelligence 35 (8) (2013) 1798–1828.

[85] 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.

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

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

[88] 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, Journal of machine learning research 11 (Dec) (2010) 3371–3408.

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

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

[91] G. E. Hinton, A better way to learn features: technical perspective, Communications of the ACM 54 (10) (2011) 94–94.

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

[93] 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.

[94] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324. doi:10.1109/5.726791.

[95] 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, NIPS'12, Curran Associates Inc., USA, 2012, pp. 1097–1105.
URL http://dl.acm.org/citation.cfm?id=2999134.2999257

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

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

[98] A. Sherstinsky, Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network, CoRR abs/1808.03314.

[99] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–80. doi:10.1162/neco.1997.9.8.1735.

[100] 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 vol.3. doi:10.1109/IJCNN.2000.861302.

[101] J. Chung, aglar Gülehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, CoRR abs/1412.3555.

[102] H. Sak, A. W. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: INTERSPEECH, 2014.

[103] 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.

[104] 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 Transactions on Audio, Speech, and Language Processing 24 (2016) 694–707.

[105] 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.

[106] 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, NIPS'16, Curran Associates Inc., USA, 2016, pp. 2234–2242.
URL `http://dl.acm.org/citation.cfm?id=3157096.3157346`

[107] R. Groß, Y. Gu, W. Li, M. Gauci, Generalizing gans: A turing perspective, in: NIPS, 2017.

[108] 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.

[109] C. Vondrick, H. Pirsiavash, A. Torralba, Generating videos with scene dynamics, in: NIPS, 2016.

[110] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, H. Lee, Generative adversarial text to image synthesis, in: ICML, 2016.

[111] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Im-

ageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (3) (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[112] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, CoRR abs/1311.2901. arXiv:1311.2901.
URL http://arxiv.org/abs/1311.2901

[113] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CoRR abs/1409.4842. arXiv:1409.4842.
URL http://arxiv.org/abs/1409.4842

[114] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556. arXiv:1409.1556.
URL http://arxiv.org/abs/1409.1556

[115] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR abs/1512.03385. arXiv:1512.03385.
URL http://arxiv.org/abs/1512.03385

[116] L. N. Smith, A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay, CoRR abs/1803.09820. arXiv:1803.09820.
URL http://arxiv.org/abs/1803.09820

[117] F. Assunão, N. Loureno, P. Machado, B. Ribeiro, Denser: deep evolutionary network structured representation, Genetic Programming and Evolvable Machines (2018) 1–31.

[118] B. A. Garro, R. A. Vázquez, Designing artificial neural networks using particle swarm optimization algorithms, in: Comp. Int. and Neurosc., 2015.

[119] G. Das, P. K. Pattnaik, S. K. Padhy, Artificial neural network trained by particle swarm optimization for non-linear channel equal-

ization, Expert Systems with Applications 41 (7) (2014) 3491 – 3496. doi:https://doi.org/10.1016/j.eswa.2013.10.053.
URL http://www.sciencedirect.com/science/article/pii/S0957417413008701

[120] B. Wang, Y. Sun, B. Xue, M. Zhang, Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification, 2018 IEEE Congress on Evolutionary Computation (CEC) (2018) 1–8.

[121] S. Sengupta, S. Basak, R. A. Peters, Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives, Machine Learning and Knowledge Extraction 1 (1) (2018) 157–191. doi:10.3390/make1010010.
URL http://www.mdpi.com/2504-4990/1/1/10

[122] 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. doi:10.1109/SSCI.2018.8628792.

[123] S. Sengupta, S. Basak, R. A. Peters, Chaotic quantum double delta swarm algorithm using chebyshev maps: Theoretical foundations, performance analyses and convergence issues, Journal of Sensor and Actuator Networks 8 (1). doi:10.3390/jsan8010009.
URL http://www.mdpi.com/2224-2708/8/1/9

[124] M. Hüttenrauch, A. Sosic, G. Neumann, Deep reinforcement learning for swarm systems, CoRR abs/1807.06613.

[125] 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. doi:10.1109/TEST.1995.529902.

[126] T. M. Khoshgoftaar, R. M. Szabo, Using neural networks to predict software faults during testing, IEEE Transactions on Reliability 45 (3) (1996) 456–462. doi:10.1109/24.537016.

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

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

[129] 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.

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

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

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

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

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

[135] A. Pumsirirat, L. Yan, Credit card fraud detection using deep learning based on auto-encoder and restricted boltzmann machine, INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS 9 (1) (2018) 18–25.

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

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

[138] 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 Networks 102 (2018) 78 – 86. doi:https://doi.org/10.1016/j.neunet.2018.02.015.
URL http://www.sciencedirect.com/science/article/pii/S0893608018300698

[139] M. Dong, L. Yao, X. Wang, B. Benatallah, C. Huang, X. Ning, Opinion fraud detection via neural autoencoder decision forest, Pattern Recognition Lettersdoi:https://doi.org/10.1016/j.patrec.2018.07.013.
URL http://www.sciencedirect.com/science/article/pii/S0167865518303039

[140] J. A. Gmez, J. Arvalo, R. Paredes, J. Nin, End-to-end neural network architecture for fraud scoring in card payments, Pattern Recognition Letters 105 (2018) 175 – 181, machine Learning and Applications in Artificial Intelligence. doi:https://doi.org/10.1016/j.patrec.2017.08.024.
URL http://www.sciencedirect.com/science/article/pii/S016786551730291X

[141] 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, Engineering Applications of Artificial Intelligence 76 (2018) 130 – 157. doi:https://doi.org/10.1016/j.engappai.2018.07.008.

URL http://www.sciencedirect.com/science/article/pii/S0952197618301520

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

[143] 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 Systems with Applications 55 (2016) 194–211.

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

[145] E. F. Fama, Random walks in stock market prices, Financial analysts journal 51 (1) (1995) 75–80.

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

[147] M. Tk, R. Verner, Artificial neural networks in business: Two decades of research, Applied Soft Computing 38 (2016) 788 – 804. doi:https://doi.org/10.1016/j.asoc.2015.09.040.

URL http://www.sciencedirect.com/science/article/pii/S1568494615006122

[148] 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, Journal of King Saud University - Computer and Information Sciencesdoi:https://doi.org/10.1016/j.jksuci.2018.02.010. URL http://www.sciencedirect.com/science/article/pii/S1319157817303816

[149] 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.

[150] 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.

[151] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, European Journal of Operational Research 270 (2) (2018) 654–669.

[152] 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.

[153] 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.

[154] 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.

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

[156] S. Basak, S. Sengupta, A. Dubey, Mechanisms for Integrated Feature Normalization and Remaining Useful Life Estimation Using LSTMs Applied to Hard-Disks, arXiv e-prints (2018) arXiv:1810.08985`arXiv:1810.08985`.

[157] P. Tamilselvan, P. Wang, Failure diagnosis using deep belief learning based health state classification, Reliability Engineering System Safety 115 (2013) 124 – 135. `doi:https://doi.org/10.1016/j.ress.2013.02.022`.
URL `http://www.sciencedirect.com/science/article/pii/S0951832013000574`

[158] 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, advanced Intelligent Computing Theories and Methodologies. `doi:https://doi.org/10.1016/j.neucom.2013.03.047`.
URL `http://www.sciencedirect.com/science/article/pii/S0925231213007388`

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

[160] 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, CoRR abs/1709.01073.

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

[162] 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, KDD '16, ACM, New York, NY, USA, 2016, pp. 39–48. `doi:10.1145/2939672.2939699`.
URL `http://doi.acm.org/10.1145/2939672.2939699`

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

[164] G. van Tulder, M. de Bruijne, Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines, IEEE Transactions on Medical Imaging 35 (5) (2016) 1262–1272. `doi:10.1109/TMI.2016.2526687`.

[165] 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.

[166] 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–.
URL `http://dx.doi.org/10.1038/nature21056`

[167] 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–e4568, 29682411[pmid]. doi:10.7717/peerj.4568.
URL https://www.ncbi.nlm.nih.gov/pubmed/29682411

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

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

[170] 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.

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

[172] S. Miao, Z. J. Wang, R. Liao, A cnn regression approach for real-time 2d/3d registration, IEEE Transactions on Medical Imaging 35 (5) (2016) 1352–1363. doi:10.1109/TMI.2016.2521800.

[173] 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.

[174] V. Golkov, A. Dosovitskiy, J. I. Sperl, M. I. Menzel, M. Czisch, P. Smann, T. Brox, D. Cremers, q-space deep learning: Twelve-fold shorter and

model-free diffusion mri scans, IEEE Transactions on Medical Imaging 35 (5) (2016) 1344–1351. doi:10.1109/TMI.2016.2551324.

[175] V. S. S. Vankayala, N. D. Rao, Artificial neural networks and their applications to power systemsa bibliographical survey, Electric power systems research 28 (1) (1993) 67–79.

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

[177] 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.

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

[179] 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, International Journal of Forecasting 2 (2) (1986) 241–242.
URL https://EconPapers.repec.org/RePEc:eee:intfor:v:2:y:1986:i:2:p:241-242

[180] H. S. Hippert, C. E. Pedreira, R. C. Souza, Neural networks for short-term load forecasting: A review and evaluation, IEEE Transactions on power systems 16 (1) (2001) 44–55.

[181] C. Kuster, Y. Rezgui, M. Mourshed, Electrical load forecasting models: A critical systematic review, Sustainable cities and society 35 (2017) 257–270.

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

[183] Y. Zhai, Time series forecasting competition among three sophisticated paradigms, Ph.D. thesis, University of North Carolina at Wilmington (2005).

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

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

[186] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, J. He, Short-term load forecasting with deep residual networks, IEEE Transactions on Smart Grid.

[187] 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.

[188] 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.

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

[190] 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 Transactions on Smart Grid.

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

[192] S. A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renewable and sustainable energy reviews 5 (4) (2001) 373–401.

[193] 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 Conversion and Management 153 (2017) 409–422.

[194] 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, Renewable and Sustainable Energy Reviews 81 (2018) 912–928.

[195] V. Dabra, K. K. Paliwal, P. Sharma, N. Kumar, Optimization of photovoltaic power system: a comparative study, Protection and Control of Modern Power Systems 2 (1) (2017) 3.

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

[197] 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.

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

[199] M. Abdel-Nasser, K. Mahmoud, Accurate photovoltaic power forecasting models using deep lstm-rnn, Neural Computing and Applications (2017) 1–14.

[200] J. F. Manwell, J. G. McGowan, A. L. Rogers, Wind energy explained: theory, design and application, John Wiley & Sons, 2010.

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

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

[203] 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, Applied energy 188 (2017) 56–70.

[204] 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.

[205] C. Feng, M. Cui, B.-M. Hodge, J. Zhang, A data-driven multi-model methodology with deep feature selection for short-term wind forecasting, Applied Energy 190 (2017) 1245–1257.

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

[207] 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, CoRR abs/1702.05747. `arXiv:1702.05747`.
URL `http://arxiv.org/abs/1702.05747`

[208] D. Shen, G. Wu, H.-I. Suk, Deep learning in medical image analysis, Annual Review of Biomedical Engineering

19 (1) (2017) 221–248, pMID: 28301734. arXiv:https://doi.org/10.1146/annurev-bioeng-071516-044442, doi:10.1146/annurev-bioeng-071516-044442.
URL https://doi.org/10.1146/annurev-bioeng-071516-044442

[209] 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) (2016) 2818–2826.

[210] 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. doi:10.1109/CVPR.2012.6247908.

[211] Brain mri image segmentation using stacked denoising autoencoders, https://goo.gl/tpnDx3.

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

[213] G. Marcus, Deep learning: A critical appraisal, CoRR abs/1801.00631.

[214] S. Sabour, N. Frosst, G. E. Hinton, Dynamic routing between capsules, in: NIPS 2017, 2017.

[215] G. E. Hinton, A. Krizhevsky, S. D. Wang, Transforming auto-encoders, in: ICANN, 2011.

[216] 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, NIPS'16, Curran Associates Inc., USA, 2016, pp. 3637–3645.
URL http://dl.acm.org/citation.cfm?id=3157382.3157504

[217] K. Hsu, S. Levine, C. Finn, Unsupervised learning via meta-learning, CoRR abs/1810.02334.

[218] 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.

[219] B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning, CoRR abs/1611.02167.

[220] C. J. C. H. Watkins, P. Dayan, Q-learning, Machine Learning 8 (3) (1992) 279–292. doi:10.1007/BF00992698.
URL https://doi.org/10.1007/BF00992698