A Brief Survey of Machine Learning Methods and their Sensor and IoT Applications

Uday Shankar Shanthamallu, Andreas Spanias, Cihan Tepedelenlioglu, and Mike Stanley*
SenSIP Center, School of ECEE, Arizona State University, NXP Semiconductors*
Tempe, AZ 85287, USA
sensip@asu.edu

***Abstract*—This paper provides a brief survey of the basic concepts and algorithms used for Machine Learning and its applications. We begin with a broader definition of machine learning and then introduce various learning modalities including supervised and unsupervised methods and deep learning paradigms. In the rest of the paper, we discuss applications of machine learning algorithms in various fields including pattern recognition, sensor networks, anomaly detection, Internet of Things (IoT) and health monitoring. In the final sections, we present some of the software tools and an extensive bibliography.**

I. INTRODUCTION

Machine Learning [1-10,89], as described by Arthur Samuel in 1959 [11], is a "Field of study that gives computers the ability to learn without being explicitly programmed." In 1997, Tom Mitchell [12] gave a more formal definition, namely: "A Computer program is said to learn from an experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E."

Although, the term machine learning has its origins in computer science, there have been several vector quantization methods [106] developed in telecommunications and signal processing for coding and compression [105]. In computer and data science, learning is accomplished based on examples (data samples) and experience. A basic signal/data processing [86-88,90] framework that includes pre-processing, noise removal and segmentation is shown in Figure 1, where, the signal is acquired from the sensor and then processed, typically in a frame-by-frame or batch mode [94]. Removal of noise and feature extraction follows next and finally the classification stage which will provide either an estimate or a decision is at the end of the process.

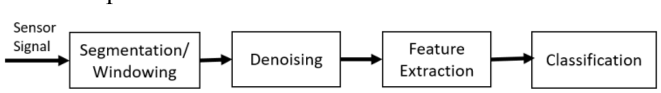

Figure 1: Basic signal processing framework including pre-processing, feature extraction and classification.

Typically, the feature extraction stage will extract compact information bearing parameters that can characterize the data. The classification stage will have to be trained by a machine learning algorithm to recognize and classify the collection of features. The field of machine learning is vast and applications are expanding rapidly especially with the emergence of fast mobile devices that also have access to cloud computing [108]. Compressing and extracting information from sensors and big data have recently elevated interest in the area. Smart city projects, mobile health monitoring, networked security, manufacturing, self-driven automobiles, surveillance, intelligent border control; every application has its idiosyncrasies and requires customized features, adaptive learning, and data fusion. Data compression and statistical signal and data analysis has a large role transmitting and interpreting data and producing meaningful analytics. Machine Learning algorithms can be broadly classified into three categories based on the properties, style of learning, and the way data are used [13]: supervised, unsupervised and semi-supervised algorithms. This type of classification is important in identifying the role of the input data, the utility of the algorithms and learning models relative to the applications.

II. SUPERVISED LEARNING

In supervised learning, "true" or "correct" labels of the input dataset are available. The algorithm is "trained" using the labeled input dataset (training data) which means ground truth samples are available for training. In the training process, the algorithm makes appropriate predictions on the input data and improves its estimates using the ground truth and reiterating until the algorithm reaches a desired level of accuracy. In almost all the machine learning algorithms, we optimize a cost function or an objective function. The cost function is typically a measure of the error between the ground truth and the algorithm estimates. By minimizing the cost function, we train our model to produce estimates that are close to the correct values (ground truth). Minimization of the cost function is usually achieved using gradient descent technique [116-118,121,122]. Variants of gradient descent technique such as stochastic gradient descent for a minibatch, momentum based gradient descent [123,124], nesterov accelerated gradient descent [119] have been used in many machine learning training paradigms. Suppose we have 'm' number of training examples, each one of them is a labelled data and can be represented in a pair:$(\boldsymbol{x}, y)$, here $\boldsymbol{x}$ represents the input data and y represents the class label. The input data $\boldsymbol{x}$ can be an n dimensional, whereas each dimension corresponds to a feature or a variable. Supervised learning methods are used in various fields including the identification of phytoplankton species [14], mapping rainfall induced landslides [15], and classification of biomedical data [16]. In [91], a machine learning algorithm is integrated on an embedded sensor system for IoT applications. In the following sub-sections, we present supervised learning algorithms.

A. Linear Regression

Regression [17-19] is a statistical technique of estimating the relationship between input and output variables. It maps the input variables to a continuous function. A simple univariate linear regression [20-22, 24] model is shown in Figure 2.

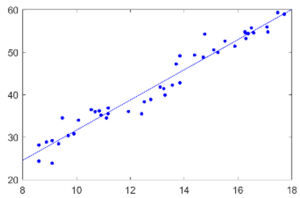

Figure 2: A simple Linear regression example with one feature/variable.

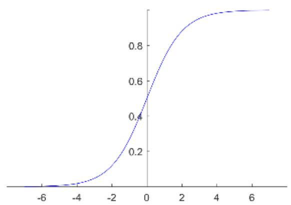

Figure 3: Sigmoid curve having a bound between 0 and 1.

The training dataset consists of 'm' labelled training sets $(\boldsymbol{x}, y) \in R^{n+1}$, $\boldsymbol{x}$ is the independent variable and y is the dependent variable. The linear regression model assumes the relationship between independent variable and dependent variable is linear and fits a straight line to the data points. This relationship is expressed by a hypothesis function or a prediction function. It is expressed as

$$h(\boldsymbol{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \qquad (1)$$

where $x_1, x_2, \ldots x_n$ are the features and $w_0, w_1, w_2 \ldots w_n$ are the weights of the model. As shown in [142] an FIR filtering approach can be used to perform linear regression through slope filtering. Equation (1) is for a multivariate linear regression model. The output is the linear sum of the weighted input features. The weights are typically learned by weighted least squares minimization process. We can also make use of quadratic, cubic or higher polynomial [144-145] terms to obtain completely different hypothesis function which can fit quadratic [143], cubic or polynomial curves respectively, rather than a simple straight line. Multivariate linear regression is used for several applications, including activity recognition and classification [23], steady state visual evoked potential (SSVEP) recognition for BCI data [25,26].

B. Logistic Regression

The objective of multivariate regression model is to determine a hypothesis function which outputs a continuous value. Now, we present another class of supervised learning algorithms: Classification, in which the objective is to obtain a discrete output. Logistic regression [30,31] is a statistical way of modelling a binomial outcome. As before, the input can have one or more features (or variables). For a binary logistic regression, the outcome can be a 0 or 1 which performs binary classification of positive class from negative class. Logistic regression uses a sigmoid curve shown in the Figure 3 to output a probability value and thus performs the classification. The hypothesis function for a logistic regression is given by

$$h(x) = S(w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n) \qquad (2)$$

where $S(\,)$ is a sigmoid function given by

$$S(z) = \frac{1}{1 + e^{-z}} \qquad (3)$$

The output of the sigmoid function is a value between 0 and 1. All values below 0.5 belong to negative class and values greater than or equal to 0.5 belong to positive class. The application of Logistic Regression is seen in various fields including evaluating Trauma care [27], patient severity assessment [28], determining the risk of heart disease [29], early detection and recognition of Glaucoma in ocular thermographs [32], and in computer vision and adaptive object tracking [33]. For a multiclass classification problem, we can have one-vs-all implementation.

C. Support Vector Machines (SVM)

Support Vector Machines [1-4,34,35,37] are one of the popular supervised learning models, mainly used for binary classification as well as multi-class classification. SVM maps the input data as points in a 'n' dimensional space and draws a '$n-1$' dimensional hyperplane to separate the data points into two groups. This can be visualized easily for two-dimensional data points as shown in the Figure 4. From the labeled dataset, SVM algorithm tries to divide these points to two separate groups by a hyperplane, which is in this case a line, such that the width of separation between the two groups is maximized. In the Figure 4, 'B' is a line which just separates two classes. However, the line 'A' gives the maximum separation between the classes. The data points which are close to the hyperplane (line 'A') are called support vectors. Maximum margin was proposed by Vapnik in 1963 and the SVM algorithm was introduced in 1992 [36]. Vapnik et.al also proposed a technique to generate a non-linear hyperplane known as the "Kernel trick" when the data is non-linearly separable. The kernel trick is achieved by transforming the non-linearly separable input data to a higher dimensional space or Hilbert space, where, the transformed data is now linearly separable. The linear hyperplane is drawn in this space and transformed back into original feature space. Many types of kernels are used in practice including Gaussian kernels [130-134], the radial basis function [120], and the polynomial kernel [125-128]. In 1995, Vapnik and Cortes proposed the soft-margin approach [38] where the maximum margin constraint is relaxed by introducing the slack variables which allows outliers of either class to be present on the other side of the hyperplane. A major advantage of SVM is that it avoids overfitting and is non-probabilistic. SVM can also be used for regression analysis as well as clustering [39-41]. The SVM algorithm is used in several applications including simple binary classification [135] text categorization [136-138], hand written digit recognition [139-141], novelty, anomaly or outlier detection [42,43], intrusion detection [51], emotion recognition [67], stress detection [69], noise robust speech recognition [129]. Different variations of SVM have also been proposed including the least square SVM (LS-SVM) [44], one-class SVM for anomaly detection [45-50, 85], and adaptive SVM [53].

D. Naïve Bayes

Naïve Bayes [68] classifiers are simple probabilistic classifiers. The term "Naïve" is used because of the strong assumption of the algorithm, that, all the input features are independent of each other and no correlation exists between them. Naïve Bayes is based on Bayes' theorem. Being a probabilistic model, Naïve Bayes' outputs a posterior probability of belonging to a class given the input features.

$$p(\omega_c|\boldsymbol{x}) = p(\omega_c|x_1, x_2, x_3, \ldots x_n) \tag{4}$$

$$p(\omega_c|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\omega_c)p(\omega_c)}{p(\boldsymbol{x})} \tag{5}$$

for each C possible outcomes or C number of classes. Here, $p(\omega_c|x)$ is the posterior probability that given feature $\boldsymbol{x}$ belongs to c^{th} class ω_c, and $p(\omega_c)$ is the prior probability of the class ω_c independent of the data, and $p(\boldsymbol{x}|\omega_c)$ is the likelihood which is the probability of the predictor given the class and $p(\boldsymbol{x})$ is the prior probability of the predictor which is the normalizing factor. There are many variations of Naïve Bayes theorem, some of them tackle the poor assumptions of Naïve Bayes [54,55,56]. Naïve Bayes algorithm is used for text classification [57], for credit scoring [58], for emotion classification and recognition [67], and detection of epileptic seizures from EEG signals [146].

E. *k-Nearest Neighbors*

The k-Nearest Neighbors (k-NN) algorithm [1,60,61,65] is one of the simplest supervised machine learning algorithm. k-NN can be used for classification of input points to discrete outcomes. A simple k-NN model is shown in Figure 5.

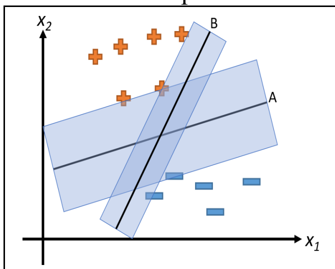

Figure 4: Maximum margin intuition; hyperplane A has maximum separation.

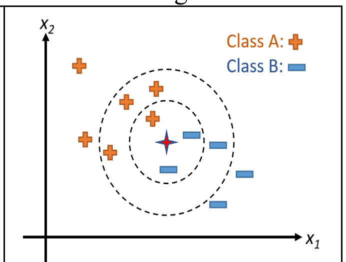

Figure 5: A simple k-NN model for different values of k

k-NN can be used for regression analysis [64,147] where the outcome of a dependent variable is predicted from the input independent variables. In Figure 5, for k =3, the test point (star) is classified as belonging to class B and for k=6, the point is classified as belonging to class A. k-NN is a non-probabilistic and non-parametric model [62,63,93] and hence it is the first choice for classification study when there is no prior knowledge about the distribution of data. k-NN stores all the labelled input points to classify any unknown sample and this makes it computationally expensive. The classification is based on the similarity measure (a distance metric). Any unknown sample is classified by the majority vote of its k nearest neighbors. The complexity increases as the dimensionality increases and hence dimensionality reduction techniques [164] are performed before using k-NN to avoid the effects of curse of dimensionality [66]. k-NN classifier is used for stress detection using physiological signals in [69] and detection of epileptic seizures [146].

III. Unsupervised Learning

In the case of unsupervised algorithms [70,71], there are no explicit labels associated with the training dataset. The objective is to draw inferences from the input data and then model the hidden or the underlying structure and the distribution in the data, in order to learn more about the data. Clustering is the most common example of an unsupervised algorithm. The details of the same is mentioned below.

A. *Clustering*

Clustering [75,81,82] deals with finding a structure or pattern in a collection of unlabeled dataset. For a given dataset, clustering algorithm groups the given data into K number of clusters such that the data points within each cluster are similar to each other and data points from different clusters are dissimilar. Similar to k-NN algorithm, we make use of a similarity metric or distance metric. Different distance metrics such as Euclidean, Mahalanobis, cosine, Minkowski etc. are used. Although Euclidean distance metric is used more often, it is shown in [74] that it is not a suitable metric to capture the quality of the clustering. The K-means algorithm is one of the simplest clustering algorithms and is an intuitive and iterative algorithm. It clusters the data by separating them into K groups of equal variances, minimizing the inertia or within-cluster sum-of-squares. However, the algorithm requires the number of clusters to be specified before running the algorithm. Each observation or the data point is assigned to the cluster with the nearest mean $\boldsymbol{\mu}_{(j)}$, which is also referred to as the Centroid of that cluster. Thus, the K clusters can be specified by the K centroids. After the random assignment of K centroids, the algorithms inner loop iterates over the following two steps:

(i) Assign each observation $\boldsymbol{x}_{(i)}$ ($\boldsymbol{x}_{(i)}$ is the i^{th} sample point) to the closest cluster centroid $\boldsymbol{\mu}_{(j)}$

(ii) Update each cluster's centroid to the mean of the points assigned to it.

$i = 1,2, \ldots m$; Total of m observations or data points

$j = 1,2, \ldots K$; Total of K clusters and hence K centroids

The inertia or the within-cluster sum-of-squares is given by:

$$\sum_{i=0}^{m} \min_{\mu_{(j)}\in C} \left\|\boldsymbol{x}_{(i)} - \boldsymbol{\mu}_{j(i)}\right\|^2 = \sum_{k=1}^{K} \sum_{i\in C} \left\|\boldsymbol{x}_{(i)} - \boldsymbol{\mu}_{(k)}\right\|^2 \tag{6}$$

$\boldsymbol{\mu}_{j(i)}$ denotes that, for the i^{th} sample, $\boldsymbol{\mu}_{\boldsymbol{j}}$ is the closest centroid. K-means clustering algorithms leads to Voronoi tessellation. K-means algorithms iterations stops (converges) when there is no change in the value of means of the clusters. In Figure 6, a converged K-means algorithm is shown. Clustering has several applications in many fields. In biology, clustering has been used to determine groups of genes that have similar functions [77-79], for detection of brain tumor in [76], cardiogram data clustering [80], in business and e-commerce analysis [83] and information retrieval [92], image segmentation [72] and compression [84], in the study of quantitative resolutions of nanoparticles [95], in fault detection in Solar PV panels [101,187,188] and in speech recognition [148].

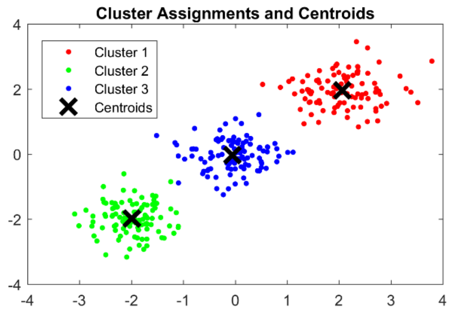

Figure 6: The K-means and the cluster centroids.

B. Vector Quantization

In its simplest form vector quantization [102,103,106] organizes data in vectors and represents them by their centroids. It typically uses a K-means clustering algorithm to train the quantizer. The centroids form codewords and all the codewords are stored in a Codebook. Vector quantization is a lossy compression method and is used in several coding applications. As a result, the compressed data has errors that are inversely proportional to density. This property is shown in Figure 8 and compared with uniform quantization Figure 7.

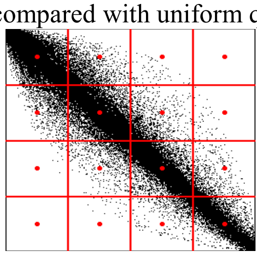

Figure 7: Uniform quantization of 2-dimensional Data.

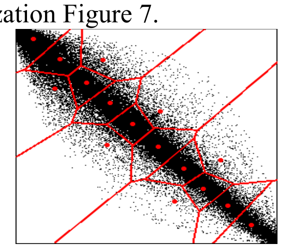

Figure 8: Vector quantization of 2-dimensional Data.

The Vector quantization technique is used in various speech applications including speech coding [103,107], emotion recognition [104], audio compression [105], large-scale image classification [149] and image compression [150].

IV. Deep Learning

In this section, a brief introduction to the field of artificial neural networks is provided with the focus on deep learning [151,153,161] methodologies and their applications. Artificial neural networks are widely used in the areas of image classification, pattern recognition and they have proved to be the most successful and they achieve superior results in various fields including signal processing [163,168,171], computer vision [157], speech processing [162,165,166] and natural language processing [158,186]. Deep learning is a branch in machine learning that has gained popularity quite recently, capable of learning multiple levels of abstraction. Although, the inception of neural networks dates in 1960 [156], deep learning gained more popularity since 2012 [155] because of the great advancements in the GPUs [99] and availability of large labeled datasets. In Figure 9, a simple artificial neural network with 4 hidden layers is shown. The last layer, namely the output layer, performs classification. The term "deep learning" [159] refers to several layers used to learn multiple levels of representation [152,154,170]. Each successive layer takes the output of the previous layer and feeds the result to the next layer.

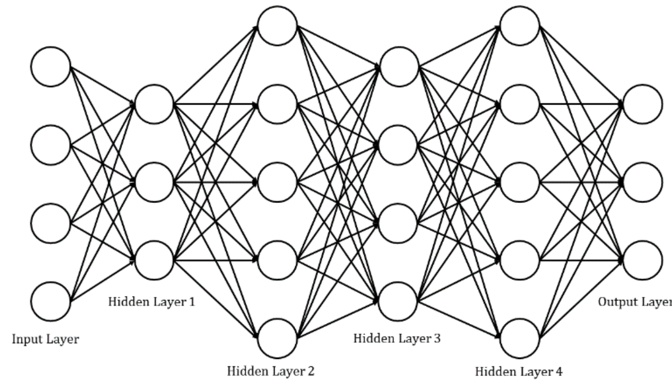

Figure 9: Artificial Neural Network with four hidden layers.

Typical artificial neural networks challenges include initialization of the network parameters, overfitting, and long training time. We now have various techniques to address the above problems. Batch normalization [182], normalization propagation [183], weight normalization [184], layer normalization [185] all help in accelerating the training of deep neural networks. Dropouts [160] help in reducing overfitting. There are several network architectures including the one shown in Figure 9 which consists of dot product layers (fully connected layers). A convolutional layer [167] processes volume of activations rather than a vector and produces *feature maps*. It also makes use of a *subsampling layer* or a *max-pooling layer* to reduce the size of the feature maps. Figure 10 shows an example of a convolutional neural network (CNN). Networks whose output depends on present and past inputs, namely recurrent neural networks (RNNs) [169,172,173], have also been used in several applications.

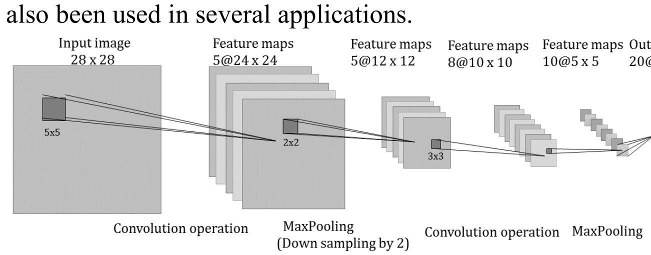

Figure 10: A CNN with 3 convolutional, 2 subsampling layers.

V. Sensor and IOT Applications

The Internet of Things (IoT) [189] is a system of connected physical devices, smart machines or objects that have unique identifiers. The devices will typically consist of electronics, software, sensors, and radios enabling these objects to continuously collect and transfer data. Sensors that consist of a transducer that will convert some form of physical process into an electrical signal. Examples include microphones, cameras, accelerometers, thermometers, pressure sensors etc. Perhaps a mobile phone is a good example of a connected device that embeds several heterogeneous sensors including microphone arrays, at least two cameras, magnetometers, accelerometers etc. First generation smart phones for example typically included six sensors. These days a Galaxy S5 has 26 sensors including microphones, cameras, magnetometer, accelerometer, proximity, IR, pressure, humidity, gyro etc. Accelerometers and magnetometers (Fig. 11) have been used in many applications, including machine monitoring, structural monitoring, human activity, and healthcare [190-193]. Other areas of collaborative sensing and machine learning include localization [199-201]

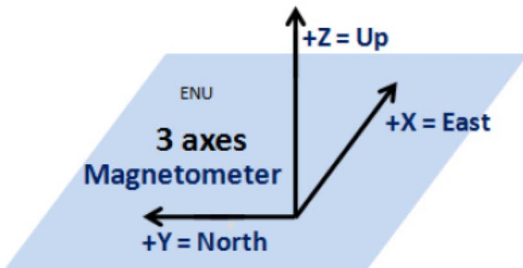

Figure 11. A magnetometer can help align with the earth's field.

Clever entertainment and information exchange systems such as smart speakers combine multiple technologies such as circular microphone arrays (Fig. 12), local and cloud based machine learning and information retrieval algorithms. The Amazon Echo represents a recent example of an IoT device that has a circular microphone array along with voice recognition capabilities. Local and cloud computing allow this device to: interface with various other systems, exchange information, provide e-services, playback music and news on demand, and provide human to machine interface for a smart home.

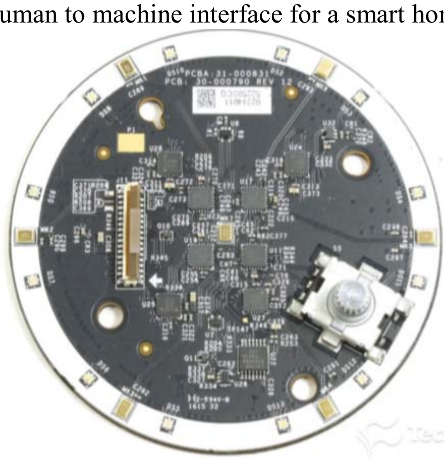

Figure 12: Microphone array on Amazon Echo™.(from [202])

The interconnection of IoT smart devices is also enabling advanced large-scale applications such as smart cities [194,195], large-scale smart networks and radios, smart campus systems [196-198]. The field of sensors and IoT applications is vast and large-scale applications are beginning to emerge. These include several smart and connected health and community systems.

VI. Implementation and Software Tools

This section introduces some of the machine learning tools. All the algorithms explained in sections II and III can be implemented in various platforms and libraries, e.g., the R [110,113] and Python [180] languages. Python is one of the most utilized environments for machine learning. There are also a number of libraries available such as SciKit-Learn [114,179] and NumPy [177,178]. TensorFlow [115,181] is an open source software library for numerical computation using data flow graphs and is very popular in deep learning and computer vision. The Azure Machine Learning Studio [111,112,176] is a drag and drop tool for analytics. IBM Bluemix [174,175] is a cloud platform that supports several programming languages as well as integrated DevOps.

Conclusion

This Machine Learning short survey paper supported the tutorial session of the IISA2017. The paper covered supervised and unsupervised learning models. We also provided a brief introduction to current deep learning methodologies and outlined several applications including pattern recognition, anomaly detection, computer vision, speech processing, and IoT applications. The paper provides extensive bibliography of machine algorithms and their applications.

Acknowledgment

This work was supported in part by the NSF I/UCRC award 1540040, IUSE award 1525716, NXP, and the ASU SenSIP Center.

References

[1] C. M. Bishop, Pattern recognition and machine learning (information science and statistics). New York: Springer-Verlag New York, 2008.
[2] R. O. Duda, D. G. Stork, and P. E. Hart, Pattern classification: Pt.1: Pattern classification, 2nd ed. New York: John Wiley & Sons, 2000.
[3] S. Marsland, Machine learning: An algorithmic perspective. Boca Raton: Chapman & Hall/CRC, 2009.
[4] Y. Kodratoff, Introduction to machine learning. Morgan Kaufmann, 1993.
[5] R. S. Michalski et al., Machine learning an artificial intelligence approach. Berlin, Heidelberg, Springer, 1983.
[6] J. Friedman, T. Hastie, and R. Tibshirani, The elements of statistical learning: Data mining, inference, and prediction, Springer, NY, 2009.
[7] Miroslav Kubat, An Introduction to Machine Learning. Springer International Publishing, ISBN 978-3-319-20009-5, 2015.
[8] "Machine learning," in Wikipedia, Wikimedia Foundation, 2016. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning.
[9] E. Alpaydin, Introduction to machine learning, MIT Press, 2010.
[10] Alexander Johannes Smola and S. Vishwanathan, Introduction to Machine Learning, Cambridge University Press, 2008.
[11] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of R&D*, vol. 3, no. 3, pp. 210–229, Jul. 1959.
[12] T. M. Mitchell, Machine learning, 7th ed. NY, McGraw Hill, 1997.
[13] J. L. Berral-García, "A quick view on current techniques and machine learning algorithms for big data analytics," *ICTON* Trento, pp.1-4, 2016.
[14] T. Phan et al., "Comparative study on supervised learning methods for identifying phytoplankton species," *ICCE*, Vietnam, pp. 283-288, 2016.
[15] S. Heleno, M. Silveira, M. Matias and P. Pina, "Assessment of supervised methods for mapping rainfall induced landslides in VHR images," *IGARSS*, Milan, pp. 850-853, 2015.
[16] P. Drotar and Z. Smeakal, "Comparative Study of Machine Learning Techniques for Supervised classification of Biomedical Data,"Acta Electrotechnica et Informatica, vol. 14, pp. 5–10, Sep. 2014.
[17] F. Galton, Natural Inheritance, *Proc Royal Soc.y of London*, 1989.
[18] F. Galton, Anthropological Miscellanea: "Regression towards mediocrity in hereditary stature,", *The Journal of the Anthropological Institute of Great Britain and Ireland*, pp. 246–263, 1886.
[19] F.Galton, "Co-relations and their measurement, chiefly from anthropometric data, Proc.s Royal Soc. of London, pp. 135-145, 1989.
[20] G. A. F. Seber, A. J. Lee, and R. A. Lee, Linear regression analysis, 2nd ed. New York, NY, United States: Wiley, John & Sons, 2003.
[21] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to linear regression analysis, 5th ed. Oxford: Wiley-Blackwell, 2012.

[22] H. Motulsky and A. Christopoulos, Fitting models to biological data using linear and nonlinear regression: A practical guide to curve fitting. New York: Oxford University Press, 2004.
[23] S. Gayathri et al., "Multivariate linear regression based activity recognition and classification," *ICICES* Chennai, pp. 1-6, 2014.
[24] P. Chandler et al., "Constrained Linear Regression for Flight Control System Failure Identification," *ACC*, San Francisco, p. 3141, 1993.
[25] H. Wang et al., "SSVEP recognition using multivariate linear regression for brain computer interface," *ICCC* Chengdu, pp. 176-180, 2015.
[26] H. Wang *et al*., "Discriminative Feature Extraction via Multivariate Linear Regression for SSVEP-Based BCI," in *IEEE Trans. on Neural Systems and Rehabilitation Eng.*, vol. 24, no. 5, pp. 532-541, May 2016.
[27] C. R. Boyd et al., "Evaluating trauma care," The Journal of Trauma: Injury, Infection, and Critical Care, vol. 27, pp. 370–378, Apr. 1987.
[28] J. R. Le Gall, "A new simplified acute physiology score (SAPS II) based on a European/north American multicenter study," *JAMA*, vol. 270, no. 24, pp. 2957–2963, Dec. 1993.
[29] J. Truett, J. Cornfield, and W. Kannel, "A multivariate analysis of the risk of coronary heart disease in Framingham," *Journal of Chronic Diseases*, vol. 20, no. 7, pp. 511–524, Jul. 1967.
[30] J. M. Hilbe, Logistic regression models. Boca Raton: Chapman and Hall/CRC, 2016.
[31] F. C. Pampel, Logistic regression: A primer. Thousand Oaks, CA: Sage Publications, 2000.
[32] Harshvardhan G et al., "Assessment of Glaucoma with ocular thermal images using GLCM techniques and Logistic Regression classifier," *WiSPNET*, Chennai, India, pp. 1534-1537, 2016.
[33] J. Song and B. Fan, "Adaptive object tracking with logistic regression," *CCDC*, Yinchuan, pp. 5403-5408, 2016.
[34] N. Cristianini et al., An introduction to support vector machines: And other kernel-based learning methods. Cambridge University Press, 2000.
[35] I. Steinwart and A. Christmann, Support vector machines. New York: Springer-Verlag New York, 2008.
[36] Boser, B. E et al., "A training algorithm for optimal margin classifiers". Proceedings of the fifth annual workshop on COLT, p. 144, 1992.
[37] C. J. C. Burges. "A Tutorial on Support Vector Machines for Pattern" Recognition. Knowledge Discovery and Data Mining, 2(2), 1998.
[38] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[39] Vapnik, Vladimir et al., "Support vector clustering", Journal of Machine Learning Research, vol. 2, pp. 125–137 2001.
[40] A. BenHur, "Support vector clustering," Scholarpedia, 3, p.5187, 2008.
[41] H.Xiao et al., "Indicative Support Vector Clustering with Its Application on Anomaly Detection, "*ICMLA* Miami, FL, pp. 273-276, 2013.
[42] D. Huang, J. H. Lai and C. D. Wang, "Incremental support vector clustering with outlier detection," *ICPR, 2012.*
[43] F. de Morsier et al., "Unsupervised change detection via hierarchical support vector clustering," *PRRS, 2012*
[44] Suykens, J.A.K.; Vandewalle, J. "Least squares support vector machine classifiers", Neural Processing Letters, 9 (3), 293-300, 1999.
[45] Bernhard S. et.al, SVM Method for Novelty detection, MIT press, 2000.
[46] B Scholkopf et al., Single class support vector machines, Unsupervised learning, Dagstuhl –seminar report, pp. 19-20, 1999
[47] D.M.J Tax and R.P.W. Duin, Data description by support vectors. In M.Verleysen, editor, *Proceedings ESANN*, Brussels, pp. 251-256, 1999.
[48] X. Peng et al., "Efficient support vector data descriptions for novelty detection, "Neural Comp. and App., vol.21, pp. 2023–2032, May 2011.
[49] S. Wang et al., "A modified support vector data description based novelty detection approach for machinery components," Applied Soft Computing, vol. 13, no. 2, pp. 1193–1205, Feb. 2013.
[50] M. Yao, H. Wang, "One-Class Support Vector Machine for Functional Data Novelty Detection," *3rd Cong. Int. Sys.*, Wuhan, p. 172, 2012.
[51] ZHOU Guangping, "The study of the application in intrusion detection based on SVM," Journal of Conv. Inf. Tech., vol. 8, p. 11, Mar. 2013.
[52] N. Chand et al., "A comparative analysis of SVM and its stacking with other classification algorithms," *ICACCA,* Dehradun, pp. 1-6, 2016.
[53] M. A. Oskoei et al, "Adaptive schemes applied to online SVM for BCI data classification," *IEEE EMBS*, Minneapolis, pp. 2600-2603, 2009.
[54] S. J. Russell and P. Norvig, Artificial intelligence: A modern approach. United Kingdom: Prentice Hall, 1994.
[55] Irina Rish, An empirical study of the naive Bayes classifier*, IJCAI Workshop on Empirical Methods in AI,* 2001.
[56] Rennie, J.; Shih, L.; Teevan, J.; Karger, D.; Tackling the poor assumptions of Naive Bayes classifiers, *ICML,* 2003.
[57] Chai, K.; H. T. Hn, H. L. Chieu; "Bayesian Online Classifiers for Text Classification and Filtering", *ACM SIGIR*, pp 97-104, August 2002.
[58] R. Vedala et al., "An application of Naive Bayes classification for credit scoring in e-lending platform," *ICDSE,* pp. 81-84, 2012.
[59] Lewis, D. D, Naive (Bayes) at forty: The independence assumption in information retrieval. *Proceedings of ECML*, 1998.
[60] S. Inc, "K-nearest neighbors," 2016. [Online]. Available: http://www.statsoft.com/textbook/k-nearest-neighbors.
[61] T. M. Cover and P. E. Hart, "Nearest neighbour pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21-27, Jan. 1967.
[62] L. Peterson, "K-nearest neighbor," Scholarpedia, vol. 4, p. 1883, 2009.
[63] Y. Lifshits, "Nearest neighbor search," SIGSPATIAL, v. 2, p. 12, 2010.
[64] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The Amer. Stat.*, vol. 46, p. 175, 1992.
[65] Cover TM, Hart PE, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
[66] K. Beyer et al., "When Is 'Nearest Neighbor' Meaningful*?" Database Theory: ICDT,* pp. 217–235, 1999.
[67] E. H. Jang, B. J. Park, S. H. Kim, Y. Eum and J. H. Sohn, "A Study on Analysis of Bio-Signals for Basic Emotions Classification: Recognition Using Machine Learning Algorithms," *2014 ICISA*, pp. 1-4, Seoul, 2014.
[68] W. Wu et al., "Bayesian Machine Learning: EEG/MEG signal processing measurements," in *IEEE SPM*, vol. 33, no. 1, pp. 14-36, Jan. 2016.
[69] A. Ghaderi et al., "Machine learning-based signal processing using physiological signals for stress detection," *ICBME*, Tehran, 2015.
[70] M. Khanum et al., "A survey on Unsupervised machine learning Algorithms for automation, classification and maintenance," *IJCA,* vol. 119, no. 13, pp. 34–39, Jun. 2015.
[71] M. E. Celebi , K. Aydin, Ed., Unsupervised Learning Algorithms, 1st ed. Switzerland: Springer International Publishing, 2016.
[72] A. Albiol et al., "An unsupervised color image segmentation algorithm for face detection applications," *ICIP*, Thessaloniki, pp. 681-684, 2001.
[73] C. K. Lee, P. F. Sum and K. S. Tan, "An unsupervised learning algorithm for character recognition," *Neural Networks, 1992. IJCNN.,*
[74] N. Bouhmala, "How Good is the Euclidean Distance Metric for the Clustering Problem,"*IIAI-AAI*, Kumamoto, pp. 312-315, 2016.
[75] A. Bindal and A. Pathak, "A survey on k-means clustering and web-text mining," IJSR, vol. 5, no. 4, pp. 1049–1052, Apr. 2016.
[76] A. A. Mandwe and A. Anjum, "Detection of brain tumor using k-means clustering," *IJSR*, vol. 5, no. 6, pp. 420–423, Jun. 2016.
[77] K. Dhiraj and S. K. Rath, "Gene expression analysis using clustering," Int. Journal of Comp and Elec Engg pp. 155–164, 2009.
[78] A. Bhattacharya, R. De, "Bi-correlation clustering algorithm for determining a set of co-regulated genes," Bioinf., V, 25, p. 2795, 2009.
[79] E. Zeng, C. Yang, T. Li, and G. Narasimhan, "Clustering genes using heterogeneous data sources," *IJKDB*, vol. 1, no. 2, pp. 12–28, 2010.
[80] C. Sundar, "An analysis on the performance of k-means clustering algorithm for Cardiotocogram clustering," IJCSA, v. 2, p. 11, Oct. 2012.
[81] J. Sun, "Clustering Algorithms research," J. Software, V. 19, Jun. 2008.
[82] G. Gan, Chaoqun, and J. Wu, Data clustering: Theory, algorithms, and applications. Philadelphia, SIAM, U.S., 2007.
[83] X. HUANG and Z. Song, "Clustering analysis on e-commerce transaction based on k-means clustering," J. Networks, vol. 9, Feb. 2014.
[84] C. W. Wang and J. H. Jeng, "Image compression using PCA with clustering," *ISPACS* New Taipei, pp. 458-462, 2012.
[85] Kunlun Li and Guifa Teng, "Unsupervised SVM Based on p-kernels for Anomaly Detection," *ICICIC* Beijing, pp. 59-62, 2006.
[86] N. Kovvali, M. Banavar, A. Spanias An Introduction to Kalman Filtering with MATLAB Examples, Synthesis Lect. Signal Proc., Morgan & Claypool Publ., Ed. J. Mura, vol. 6, , Sep. 2013.
[87] B. Widrow, S. Stearns, Adaptive Signal Processing, Prentice Hall, 1985.
[88] J. Foutz, A. Spanias, M. Banavar, Narrowband Direction of Arrival Estimation for Antenna Arrays, Synthesis Lectures on Antennas, Morgan & Claypool Publishers, ISBN-13: 978-1598296501, Aug. 2008.
[89] S. Theodorides, *Machine Learning, A Bayesian and Optimization Perspective*, 1st Edition, Academic Press, December 2015.
[90] A. Spanias, Digital Signal Processing; An Interactive Approach – 2nd Edition, ISBN 978-1-4675-9892-7, Lulu Press, May 2014.

[91] J. Lee, M. Stanley, A. Spanias, and Cihan Tepedelenlioglu, "Integrating Machine Learning in Embedded Sensor Systems for Internet-of-Things Applications," IEEE ISSPIT, Limassol, Cyprus Dec. 2016.

[92] J. Thiagarajan, K. Ramamurthy, P. Turaga, A. Spanias, Image Understanding Using Sparse Representations, Synth. Lect. on Image, Video, and Multimedia Proc., Morgan & Claypool Publ., April 2014.

[93] V. Berisha, A. Wisler, A. Hero, A. Spanias, "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure," *IEEE Trans. on Signal Processing*, vol. 64, pp.580-591, Feb. 2016.

[94] Wichern, G.; Jiachen Xue; Thornburg, H.; Mechtley+, B.; Spanias, A; "Segmentation, Indexing, and Retrieval for Environmental and Natural Sounds IEEE Trans on ASLP, Vol. 18, Issue: 3, pp. 688 – 707, 2010.

[95] X Bi, S Lee, JF Ranville, P Sattigeri, A Spanias, P Herckes, P Westerhoff, "Quantitative resolution of nanoparticle sizes using single particle inductively coupled plasma mass spectrometry with the K-means clustering algorithm," *J Analy. Atomic Spectr.*, 29, p. 1630, 2014.

[96] Braun, H. Turaga, P.; Spanias, A., Direct tracking from compressive imagers: A proof of concept," *IEEE ICASSP* 2014, Florence, 2014.

[97] J. J. Thiagarajan, K. N. Ramamurthy, P. Sattigeri and A. Spanias, "Supervised local sparse coding of sub-image features for image retrieval," *IEEE ICIP 2012*, Orlando, Sept. 2012.

[98] P. Sattigeri, J. J. Thiagarajan, M. Shah, K. N. Ramamurthy and A. Spanias, "A scalable feature learning and tag prediction framework for natural environment sounds," *48th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 1779-1783, 2014.

[99] P. Sattigeri, J. J. Thiagarajan, K. N. Ramamurthy and A. Spanias, "Implementation of a fast image coding and retrieval system using a GPU," *2012 IEEE ESPA*, Las Vegas, NV, pp. 5-8, 2012.

[100] P. Sattigeri, J. J. Thiagarajan, K. Natesan Ramamurthy, A. Spanias, M. Goryll and T. Thornton, "Robust PSD Features for Ion-Channel Signals," in SSPD, London, UK, 27-29 September 2011.

[101] A. Spanias, C. Tepedelenlioglu, E.Kyriakides, D. Ramirez, S. Rao, H. Braun, J. Lee, D. Srinivasan, J. Frye, S. Koizumi, Y. Morimoto, "An 18 kW Solar Array Research Facility for Fault Detection Experiments," *Proc. 18th MELECON*, Cyprus, April 2016.

[102] A. Gersho and R. M. Gray, Vector quantization and signal compression, 6th ed. Boston, MA, United States: Kluwer Academic Publishers, 1991.

[103] J. Makhoul et al., "Vector quantization in speech coding," in *Proceedings of the IEEE*, vol. 73, no. 11, pp. 1551-1588, Nov. 1985.

[104] M. Shah, C. Chakrabarti and A. Spanias, "Within and cross-corpus speech emotion recognition using latent topic model-based features", EURASIP J . Audio, Speech, and Music Processing, 2015:4, Jan. 2015.

[105] A.Spanias, T. Painter, V.Atti, Audio Signal Processing and Coding, Wiley, March 2007.

[106] Linde Yoseph Buzo Andrés Gray Robert M. "An Algorithm for Vector Quantization" *IEEE COM*-28 No. 1 pp. 84-95 Jan. 1980.

[107] A.S. Spanias, "Speech Coding: A Tutorial Review," Proceedings of the IEEE, Vol. 82, No. 10, pp. 1441-1582, October 1994.

[108] E. G. Ularu et al., "Mobile computing and cloud maturity - introducing machine learning for ERP configuration automation," Informatica Economica, vol. 17, no. 1/2013, pp. 40–52, Mar. 2013.

[109] I. H. Witten et al., "Data mining: Practical machine learning tools and techniques", 3rd ed. USA, Morgan Kaufmann Publishers In, 2011.

[110] R. Schumacker, Understanding Statistics Using R, S. Tomek, Ed. Springer Publishing Company, Inc., 2013.

[111] G. Webber-Cross, Learning Microsoft azure: A comprehensive guide to cloud application development using MS azure. UK: Packt Publi., 2014.

[112] V. Fontamaet al., "Predictive Analytics with MS azure machine learning; build and deploy solutions in minutes.USA press, 2014.

[113] R Development Core Team (2008). R: A language and environment for statistical computing. R foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

[114] Pedregosa, F et al., Scikit-learn: Machine Learning in Python, Journal on Machine Learning Research 12, p. 2825, 2011. https://scikit-learn.org

[115] Abadi, M et al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://tensorflow.org

[116] S. Amari, "Backpropagation and stochastic gradient descent method", Neurocomputing, vol. 5, no. 4-5, pp. 185-196, 1993.

[117] H. Blockeel, Machine learning and knowledge discovery in databases, Berlin, Springer, 2013.

[118] V. J. Mathews, Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," *IEEE Trans. SP*, vol. 41, p. 2075, Jun 1993.

[119] G. Qu and N. Li, "Accelerated Distributed Nesterov Gradient Descent for smooth and strongly convex functions," *54th Annual Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, pp. 209-216, 2016.

[120] Y. Wong, "How Gaussian radial basis functions work," *IJCNN-91 Int. Joint Conf. on Neural Networks*, Seattle, WA, pp. 133-138, 1991.

[121] J. A. Flanagan and T. Novosad, "Maximizing WCDMA network packet traffic performance: multi-parameter optimization by gradient descent minimization of a cost function," *IEEE PIMRC*, v..1, pp. 311-315, 2003.

[122] F. F. Lubis et al.,"Gradient descent and normal equations on cost function minimization for online predictive using linear regression with multiple variables," *2014 ICISS*, Bandung, pp. 202-205, 2014.

[123] S. K. Lenka et al., "Gradient Descent with Momentum based Neural Network Pattern Classification for the Prediction of Soil Moisture Content in Precision Agriculture," *2015 IEEE iNIS.*, Indore, p. 63, 2015.

[124] M. Tivnan et al., "A modified gradient descent reconstruction algorithm for breast cancer detection using Microwave Radar and Digital Breast Tomosynthesis," *2016 10th EuCAP*, Davos, pp. 1-4, 2016.

[125] D. Chen, et al., "Similarity learning on an explicit kernel feature map for person re-identification," *IEEE CVPR*, Boston, pp. 1565, 2015.

[126] P. Sahoo et al., "On the study of GRBF and polynomial kernel based support vector machine in web logs," *2013 1st Int. Conf. on Emerging Trends and Applications in Computer Science*, Shillong, 2013, pp. 1-5.

[127] P. Panavaranan and Y. Wongsawat, "EEG-based pain estimation via fuzzy logic and polynomial kernel support vector machine," *The 6th 2013 Biomedical Engg. Int. Conf.*, Amphur Muang, 2013, pp. 1-4.

[128] S. Yaman et al., "Using Polynomial Kernel SVM for Speaker Verification," in *IEEE Sig.Process. Lett.*, v. 20, pp. 901-904, Sept. 2013.

[129] J. Bai et al., "Application of SVM with Modified Gaussian Kernel in A Noise-Robust Speech Recognition System," *IEEE Int. Symp. on Knowledge Acquisition and Modeling Workshop*, pp. 502-505, 2008.

[130] P. Baldi; S. Brunak, "Gaussian Processes, Kernel Methods, and SVM," Bioinformatics, Machine Learning, *MIT Press*, p.387, 2001

[131] M. Varewyck et al., "A Practical Approach to Model Selection for Support Vector Machines With a Gaussian Kernel," in *IEEE Trans. on SMC, Part B (Cybernetics)*, vol. 41, no. 2, pp. 330-340, April 2011.

[132] D. Zhang et al., "Time Series Classification Using SVM with Gaussian Elastic Metric Kernel," *ICPR*, Istanbul, 2010, pp. 29-32.

[133] J. Tian and L. Zhao, "Weighted Gaussian Kernel with Multiple Widths and Support Vector Classifications," *Int. Symposium on Info. Engg. and Electronic Commerce*, Ternopil, pp. 379-382, 2009.

[134] Yaohua Tang et al., "Efficient model selection for SVM with Gaussian kernel function," *2009 IEEE CIDM*, Nashville, TN, pp. 40-45, 2009.

[135] A. Betancourt et al., "Filtering SVM frame-by-frame binary classification in a detection framework," *ICIP*, Quebec, p. 2552, 2015.

[136] Chen donghui and Liu zhijing, "A new text categorization method based on HMM and SVM," *2010 2nd Int. Conf. on Computer Engig. and Tech.*, Chengdu, 2010, pp. V7-383-V7-386.

[137] M. Kumar, M. Gopal, "An Investigation on Linear SVM and its Variants," *Int. Conf. Mach/ Learn.and Comp.*, Bangalore, p. 27, 2010.

[138] Z. Wang and X. Qian, "Text Categorization Based on LDA and SVM," *2008 Int. Conf. on Com Sci and Sofe Eng.*, Hubei, p. 674, 2008..

[139] A. Sharma, "Handwritten digit recognition using SVM", eprint arXiv:1203.3847, 2012.

[140] D. Gorgevik et al., "Handwritten digit recognition by combining support vector machines using rule-based reasoning," *Proc. 23rd Int. Conf. on Info. Tech. Interfaces,* pp. 139-144 vol.1, 2001

[141] Tuba, Eva et al. "Handwritten digit recognition by SVM optimized by bat algo." 24th Int Conf WSCG , 2016.

[142] C. S. Turner, "Slope filtering: An FIR approach to linear regression [DSP Tips&Tricks]," *IEEE Sig, Proc. Mag.,* pp. 159-163, Nov. 2008.

[143] Y. T. Chang and K. Cheng, "Sensorless position estimation of switched reluctance motor at startup using quadratic polynomial regression," in *IET Electric Power Applications*, vol. 7, pp. 618-626, Aug. 2013.

[144] E. Masry, "Multivariate regression estimation of continuous-time processes from sampled data: local polynomial fitting approach," in *IEEE Trans. on Info. Theory*, vol. 45, no. 6, pp. 1939-1953, Sep 1999.

[145] T. Banerjee et al, "PERD: Polynomial-based Event Region Detection in Wireless Sensor Networks," *2007 IEEE ICC,* pp. 3307-3312, 2007.

[146] A. Sharmila and P. Geethanjali, "DWT Based Detection of Epileptic Seizure, ," in *IEEE Access*, vol. 4, pp. 7716-7727, 2016.

[147] V. Agrawal, et al., "Application of K-NN regression for predicting coal mill related variables," *2016 ICCPCT*, India, pp. 1-9, 2016.
[148] X. Li et al., "Speech recognition based on k-means clustering and NN ensembles," *7th Int. Conf. on Natural Comp.*, Shanghai, p. 614, 2011.
[149] E. C. Ozan, et al., "A vector quantization based k-NN approach for large-scale image classification," *IPTA*, Oulu, pp. 1-6, 2016.
[150] D. Valsesia, P. Boufounos, "Multispectral image compression using vector quantization," *IEEE ITW*, , Cambridge, p 151, 2016.
[151] I. Goodfellow, Y. Bengio and A. Courville, Deep learning, 1st ed. Cambridge, Mass: The MIT Press, 2017.
[152] Y. Bengioet et all, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on PAMI*, vol. 35, p. 1798, Aug. 2013.
[153] I. Arel et al., "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," in *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 13-18, Nov. 2010.
[154] LeCun et al., "Deep learning." nature 521.7553, 2015.
[155] Krizhevsky et al, "ImageNet classification with deep convolutional NN," *Adv. Neural Info.Process. Sys.*, vol 25, pp 1090-1098, 2012.
[156] F. Rosenblatt, "The perceptron: A probabilistic model for information storage in the brain." Psychological review vol 65, pp. 386-408, 1958.
[157] Kavukcuoglu, Koray et al. "Learning convolutional feature hierarchies for visual recognition." Advances in neural info. process. sys. 2010.
[158] Mikolov, Tomas et al. "Distributed representations of words and phrases and their compositionality." *NIPS* 2013.
[159] Schmidhuber, Jürgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015): 85-117.
[160] N. Srivastava, et al. "Dropout: A simple way to prevent neural networks from overfitting." J. Machine Learning Res, 15.1 (2014): 1929-1958.
[161] L. Deng. "A tutorial survey of architectures, algorithms, and applications for deep learning." *APSIPA Trans. on Signal and Info. Process. 3,* 2014.
[162] Hinton, G et al. "Deep NN for acoustic modeling in speech recognition", *IEEE Signal Process. Magazine*, vol. 29, no. 6, pp. 82-97, 2012.
[163] Yu, Dong, and Li Deng. "Deep learning and its applications to signal an information processing." *IEEE Sig. Pro. Mag.* 28, no. 1, pp. 145-154, 2011.
[164] Hinton, G.; Salakhutdinov, R. "Reducing the dimensionality of data with neural networks", Science 313 no. 5786, pp. 504–507, 2006.
[165] Yu, Dong, and Li Deng. Automatic speech recognition: A deep learning approach. Springer, 2014.
[166] Abdel-Hamid, Ossama, and Hui Jiang. "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code." Proc. *IEEE* ICASSP 2013, Vancouver, 2013.
[167] Szegedy, Christian et al. "Going deeper with convolutions." Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition. 2015.
[168] H. Song, et al., "Auto-context modeling using multiple Kernel learning," *2016 IEEE ICIP*, pp. 1868-1872, Phoenix, Sep. 2016.
[169] Bengio, Yoshua et al., "Advances in optimizing recurrent networks." *Proc. IEEE ICASSP, 2013,* Vancouver, 2013.
[170] Salakhutdinov et al., "Deep boltzmann machines." *Proceedings of the int. conf. on AI and statistics*. vol. 5, no. , Cambridge, MIT Press, 2009.
[171] Song, Huan, J. Jayaraman, A. Spanias, "A Deep Learning Approach To Multiple Kernel Fusion." *Proc. ICASSP 2017*, New Orleans.
[172] Mikolov, T et al.: Recurrent neural network based language model, in *Proc. IEEE ICASSP, 2010*, 1045–1048.
[173] Mesnil, G et al., Investigation of RNN architectures and learning methods for spoken language understanding, Proc. Interspeech, 2013.
[174] Kobylinski, Kris et al., "Enterprise application development in the cloud with IBM Bluemix." *Proc 24th Conf Comp. Sc. Soft. Eng*. IBM. , 2014.
[175] Gheith, A et al., "IBM Bluemix Mobile Cloud Services." IBM Journal of Research and Development 60.2-3 (2016): 7-1.
[176] Klein, Scott. "Azure Machine Learning." IoT Solutions in Microsoft's Azure IoT Suite. Apress, 2017. 227-252.
[177] Walt, Stéfan van der, et al. "The NumPy array: a structure for efficient numerical computation." Comp. in Science & Eng. 13.2 (2011): 22-30.
[178] McKinney, Wes. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc.", 2012.
[179] G. Hackeling. Mastering ML with scikit-learn. Packt Publi., 2014.
[180] Van Rossum, Guido. "Python Programming Language." USENIX Annual Technical Conf. Vol. 41. 2007.
[181] Chollet, François. "Keras: Deep learning library for theano and tensorflow." URL: https://keras. io/k (2015).
[182] I. Sergey, C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv:1502.03167 (2015).
[183] Arpit D et al. "Normalization propagation: A parametric technique for removing covariate shift in deep networks." arXiv preprint, 2016.
[184] T. Salimans et al., "Weight normalization: A reparameterization to accelerate training of neural networks." Adva. Neu. Info. Sys. 2016.
[185] J. Ba et al. "Layer normalization." arXiv:1607.06450 (2016).
[186] P. Loizou and A. Spanias, "High Performance Alphabet Recognition," *IEEE Trans. on Speech and Audio,* vol. 4, pp. 439-445, Nov. 1996.
[187] S. Rao, S. Katoch, P. Turaga, A. Spanias, C. Tepedelenlioglu, R. Ayyanar, H.Braun, J. Lee, U.Shanthamallu, M. Banavar, D. Srinivasan, "A Cyber-Physical System Approach for Photovoltaic Array Monitoring and Control," *Proceedings 8th International Conference on Information, Intelligence, Systems and Applications (IEEE IISA 2017)*, Larnaca, August 2017.
[188] A. Spanias, "Solar Energy Management as an Internet of Things (IoT) Application," *Proceedings 8th International Conference on Information, Intelligence, Systems and Applications (IEEE IISA 2017)*, Larnaca, August 2017.
[189] Gubbi, Jayavardhana et al., "Internet of Things (IoT): A vision, architectural elements, and future directions. *"Future generation computer systems"*, Vol. 29, no. 7, pp.1645-1660, 2013.
[190] Aldrich and L. Auret, Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods, Springer, 2013.
[191] X. Long, B. Yin and R. M. Aarts, "Single-accelerometer-based daily physical activity classification," in Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009.
[192] D. Rajan, A. Spanias, S. Ranganath, M. Banavar, and P. Spanias, "Health Monitoring Laboratories by Interfacing Physiological Sensors to Mobile Android Devices," in IEEE FIE, 2013.
[193] J. P. Lynch, "A Summary Review of Wireless Sensors and Sensor Networks for Structural Health Monitoring," The Shock and Vibration Digest, vol. 38, no. 2, pp. 91 - 128, 2006.
[194] Hwang, Jong-Sung; Choe, Young Han (February 2013). "Smart Cities Seoul: a case study" (PDF). ITU-T Technology Watch. Retrieved 23 October 2016.
[195] Zanella, Andrea; Bui, Nicola; Castellani, Angelo; Vangelista, Lorenzo; Zorzi, Michele (February 2014). "Internet of Things for Smart Cities". IEEE Internet of Things Journal. 1(1): 22–32. Retrieved 26 June 2015.
[196] Sensor networks and the smart campus," 2014. [Online]. Available: https://beaverworks.ll.mit.edu/CMS/bw/smartcampusfuture. Accessed: Dec. 12, 2016.
[197] P. Bellavista et al, "Convergence of MANET and WSN in IoT urban scenarios," IEEE Sens. J., vol. 13, no. 10, pp. 3558–3567, Oct. 2013.
[198] Andrea Zanella et al, "Internet of Things for Smart Cities"*, IEEE Internet Of Things Journal*, Vol. 1, No. 1, February 2014.
[199] S. Miller, X. Zhang, A. Spanias, Multipath Effects in GPS Receivers, Synthesis Lectures on Communications, Morgan & Claypool Publishers, ISBN 978-1627059312, Ed. W. Tranter, No. 1 , Dec. 2015.
[200] X. Zhang, C. Tepedelenlioglu, M. Banavar, A. Spanias, Node Localization in Wireless Sensor Networks, Synth Lectures on Communications, Morgan & Claypool Publ., ISBN: 9781627054850, Ed. W. Tranter, Dec. 2016.
[201] Quoc-Huy Phan andSu-Lim Tan, "Mitigation of GPS periodic multipath using nonlinear regression," 19th European Signal Processing Conference, Barcelona, 2011.
[202] www.ifixit.com/Teardown/Amazon+Echo+Teardown/