# Integrating Machine Learning in Embedded Sensor Systems for Internet-of-Things Applications

Jongmin Lee, *Michael Stanley, Andreas Spanias, and Cihan Tepedelenlioglu
School of ECEE, SenSIP Center, Arizona State University
{jongmin.lee, spanias, cihan}@asu.edu
*NXP Semiconductors
mike.stanley@nxp.com

***Abstract*—Interpreting sensor data in Internet-of-Things applications is a challenging problem particularly in embedded systems. We consider sensor data analytics where machine learning algorithms can be fully implemented on an embedded processor/sensor board. We develop an efficient real-time realization of a Gaussian mixture model (GMM) for execution on the NXP FRDM-K64F embedded sensor board. We demonstrate the design of a customized program and data structure that generates real-time sensor features, and we show details and training/classification results for select IoT applications. The integrated hardware/software system enables real-time data analytics and continuous training and re-training of the machine learning (ML) algorithm. The real-time ML platform can accommodate several applications with lower sensor data traffic.**

***Keywords*—sensor data analytics; embedded machine learning; condition monitoring; Internet-of-Things***

## I. INTRODUCTION

Integrated sensor technologies are at the core of many applications including Internet-of-Things (IoT), diagnosis of machine condition [1, 2], human body activities [3], health monitoring [4, 5], localization [6, 7], and structural monitoring [8, 9]. Embedded sensors in IoT and other applications are often associated with exceedingly large datasets which pose storage, transmission and security/privacy problems [10, 11]. A recent research report [12] states that machine-generated data will increase up to 42%, and embedded systems will provide 10% of all digital data by 2020. Sharing of sensor systems and cloud computing resources can help cope with these problems. The ability of simple embedded sensor boards to accommodate machine learning tasks, which is demonstrated in this paper, can help cope with big data and high bit rate issues. Furthermore, parametric representations of data, obtained by embedding machine learning algorithms directly on the sensor board, enables users to obtain analytics and develop efficient IoT applications.

In this paper, we describe the design, efficient realization and full integration of a machine learning algorithm on a simple NXP sensor board. The machine learning process relies on a Gaussian mixture model (GMM) [37,47] that uses the expectation-maximization (EM) algorithm with the minimum description length (MDL) criterion. The algorithm we implemented is based on a probabilistic model generated on the NXP board that characterizes the statistical features of sensor

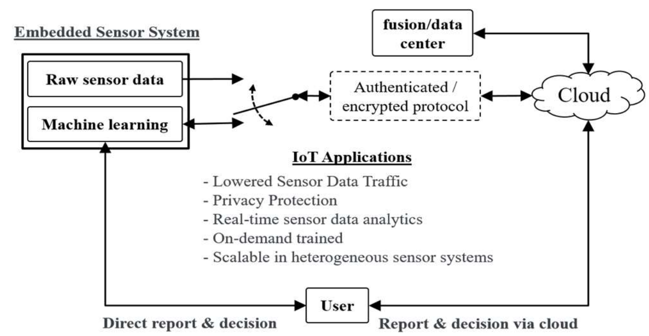



*Fig. 1. Embedded machine learning for IoT applications.*

measurements in real time. To implement the machine learning algorithm on the sensor board, we customize the data structure and computation sequence. Our implementation demonstrates the embedded machine learning and classification process with select real-time IoT experiments. We argue that the integrated sensor hardware and machine learning software has several benefits including: a) enabling the sensor system to be continuously trained and retrained, b) providing real-time analytics directly from the board, c) scalability in that the algorithms can also operate with attached heterogeneous sensors, and d) reduction of sensor data traffic and potentially enhanced privacy by communicating parameters instead of raw data (see Fig. 1).

The rest of this paper is organized as follows. Section II introduces an embedded sensor system for data analytics, and describes the steps taken to integrate a machine learning algorithm on an embedded system. In Section III, potential IoT experiments are provided. Section IV provides concluding remarks.

## II. EMBEDDED SENSORS AND MACHINE LEARNING

A key objective of the paper is to demonstrate real-time training and classification of a machine learning algorithm on an embedded sensor board for use in data analytics and IoT applications. We provide first some background information and bibliography on applications of machine learning in IoT [19, 20], mobile [24] and cloud systems [25], natural data compression [23, 26] and recognition [27,28,58,60], image classification [41], health monitoring [29,30], smart cities [21, 22], smart campus [31], transportation systems [48], solar

Research sponsored in part by NXP Semiconductors (formerly, Freescale) and the SenSIP center and NCSS I/UCRC site at Arizona State University.

panel monitoring [43,54,63] and other applications [64-68]. Machine learning algorithms are at the core of IoT systems that engage both mobile and cloud computing. Research and development is creating new technologies that have enabled new products [32, 33, 61]. Machine learning methods emerged in mobile communications in the form of vector quantization algorithms of speech features for coding and compression [23, 35, 36, 46, 49, 50]. Extensive coverage of machine learning is given in [39] and the study of Gaussian Mixture Models (GMM) was addressed in several publications [34,37,42,45, 47]. Extension, customization, and enhancement of features and machine learning methods have been discussed in [38,40, 42,57,59,62]. Kalman-type estimation methods for sensor, IoT and other systems have been covered in [44,51,52,53].

We are addressing here a sensor-based embedded system for analytics on the performance of IoT applications. We describe first the embedded sensor system which gathers raw data, obtains parameters, performs initial calculations, and provides IoT related analytics. Potential analytic scenarios were introduced in [13, 55, 56]. Manufacturers reduce the maintenance cost by proactively monitoring the performance and reliability of their products. For sensor data analytics NXP (formerly, Freescale) Semiconductors developed an embedded sensor data logger based on an Intelligent Sensing Framework (ISF). The ISF is a comprehensive and robust runtime framework providing an open sensor hub capability available for ARM-based microcontrollers and various sensors [14]. The ISF-based sensor data logger uses the FXOS8700CQ 6-axis combo accelerometer/magnetometer that is on the FRDM-K64F embedded board shown in Fig. 4. The sensor data logger collects raw sensor data in real time and records it for subsequent tasks. The existing data logger is not able to process data analytics in an embedded system but needs to transmit all the data to an IoT fusion/server center for centralized processing.

Centralized processing may increase sensor data traffic and also create vulnerabilities with data security and privacy. For example, 3-dimensional accelerometer data with high sampling frequency may generate a large set of data samples in real time. Instead, a set of model parameters obtained from data analytics can represent these data. Moreover, sensor data may contain private information that should be kept secure. Therefore, integrating data analytics in the embedded sensor board will not only reduce sensor data traffic but also enhance privacy protection. In the following sections, we implement real time a

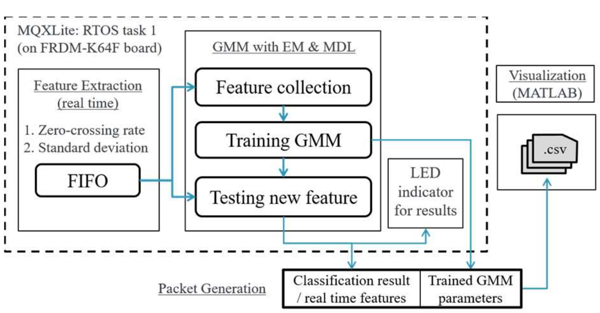



*Fig. 2. Block diagram of Embedded Machine Learning.*

GMM on an embedded sensor board. GMM is one of the most common parametric density estimation methods. We have ported the GMM, along with EM algorithm under the MDL criterion, onto the FRDM-K64F embedded board. We note that this implementation paradigm can be also ported to other embedded sensor boards. Given raw sensor data, after extracting statistical features, the implemented algorithm automatically estimates the closest distribution that is used for classification and clustering. We explain below the theoretical background followed by the implementation details.

### *A. Overview of GMM with EM and MDL*

Let $M$ be the number of Gaussian components. The $m$-th mixture model is parameterized in terms of the mean vector $\mu_m$, the covariance matrix $\Sigma_m$, and the mixing probability $p_m$. These parameters are represented for order $M$ using the notation: $\{p_m,\ \mu_m,\ \Sigma_m\}_{m=1}^{M}$. Let $\mathbf{x} \in \mathbb{R}^d$ be a real feature vector which is formed from raw sensor data. The density function of the vector $\mathbf{x}$ with $M$ components is given by

$$f(\mathbf{x};\theta,M) = \sum_{m=1}^{M} p_m g(\mathbf{x};\ \mu_m,\Sigma_m) \tag{1}$$

where the mixing probability satisfies $\sum_{m=1}^{M} p_m = 1$ and the $m$-th Gaussian component is defined as

$$g(\mathbf{x};\theta,M) = \frac{1}{(2\pi)^{d/2}|\Sigma_m|^{0.5}} e^{-\frac{1}{2}(\mathrm{x}-\mu_m)^T\Sigma_m^{-1}(\mathrm{x}-\mu_m)} \tag{2}$$

Given $N$ feature samples of $\mathbf{X} = [\mathbf{x}_1,\dots,\mathbf{x}_N]$, the parameter $\theta$ that makes GMM closely match the distribution of $\mathbf{X}$ can be achieved by maximum likelihood estimation. The likelihood function of $\theta$ for a fixed $M$, assuming independence in $\mathbf{X}$, is given by $f(\mathbf{X};\theta,M) = \prod_{n=1}^{N} f(\mathbf{x}_n;\ \theta,M)$. The maximum likelihood estimates $\hat{\theta}$ can be determined using the iterative EM algorithm [15]. With initial parameters denoted $\theta_i$ where $i = 0$, the $\theta_{i+1}$ in the next iteration is obtained by taking the expectation of $\theta_i$ with respect to $\mathbf{X}$. Each iteration estimates $f(\mathbf{X};\theta_{i+1},M)$ such that $f(\mathbf{X};\theta_{i+1},M) \geq f(\mathbf{X};\theta_i,M)$. This process is repeated until a convergence threshold is satisfied. At every iteration, the parameters of $\theta$ for the $m$-th Gaussian component are updated as shown below:

$$p_{m,i+1} = \frac{1}{N}\sum_{n=1}^{N} \Pr(m|\mathbf{x}_n,\theta_i) \tag{3}$$

$$\mu_{m,i+1} = \frac{\sum_{n=1}^{N} \Pr(m|\mathbf{x}_n,\theta_i)\,\mathbf{x}_n}{\sum_{n=1}^{N} \Pr(m|\mathbf{x}_n,\theta_i)} \tag{4}$$

$$\Sigma_{m,i+1} = \frac{\sum_{n=1}^{N} \Pr(m|\mathbf{x}_n,\theta_i)\,(\mathbf{x}_n-\mu_{m,i})(\mathbf{x}_n-\mu_{m,i})^T}{\sum_{n=1}^{N} \Pr(m|\mathbf{x}_n,\theta_i)} \tag{5}$$

The MDL criterion [16] is used to determine the Gaussian mixture model order $M$, i.e.,

$$MDL(M,\theta) = -\log f(\mathbf{X};\theta,M) + \frac{1}{2}L\log(dN) \tag{6}$$

where $d$ denotes the dimension of $\mathbf{x}$ and $L$ is the number of parameters with the model order $M$ [17]. The best positive integer order $M^*$ is determined by sequentially testing from a

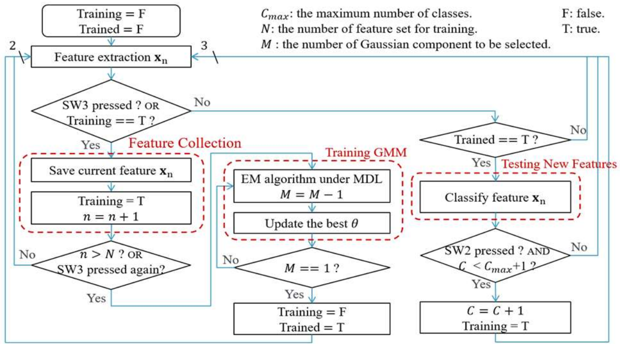



*Fig. 3. Flowchart of the GMM-based classification.*

large number of $M$ to 1. When the value of $M$ is reduced from $M$ to $M-1$, two Gaussian components are merged to a single one. The appropriate component pair is selected by finding the two closest ones such that the upper bound on the change, i.e. $MDL(M-1, \theta_{(l,k)^-}) - MDL(M, \theta)$, is minimized where $\theta_{(l,k)^-}$ are the parameters of the merging components $l, k$.

## B. Implementation

The embedded sensor board FRDM-K64F is equipped with a 32-bit ARM® Cortex-M4F core, 1MB FLASH, and 256KB RAM, along with FXOS8700CQ accelerometer. The software component was developed on top of the MQX Lite real-time operating system (RTOS) using the Intelligent Sensing Framework (ISF) library. This process can be extended to similar embedded sensor boards and RTOS.

In order for the GMM-EM algorithm to efficiently run on the FRDM-K64F with limited resources, dynamic memory allocation was not used. Statistical features were computed in fixed point representation provided by the ARM CMSIS DSP library. The implemented algorithm was partially computed in 32 bits float type using the floating-point unit of ARM Cortex M4F.We designed the data structure consisting of $\mathbf{X}$, $\theta$, and resulting classification decisions. The feature vectors are calculated from the raw data locally and used to estimate the GMM parameters $\theta$ and the order $M$ in (1). Note that $\theta$ and $M$ represent a statistical model for the sensor data and can be used to reconstruct the data-driven analytics. By performing analytics with $\theta$ and $M$, raw sensor data traffic can be reduced. Figure 2 shows the functional block diagram for the implementation on the FRDM-K64F. It collects real-time raw accelerometer data, and computes statistical features for the GMM-EM algorithm. The algorithm runs in real time within a task of the MQX Lite RTOS. Fig. 3 provides more details of the implemented algorithm. It first collects $N$ features to train a GMM. Once the GMM is trained, new features are tested in real time for classification or clustering. The implemented algorithm is controlled either by external switches on the board or by transmitting a command. The external LEDs are used as indicators for status and classification results. GMM parameters and algorithm decisions are transmitted to a host computer and visualized in real time.

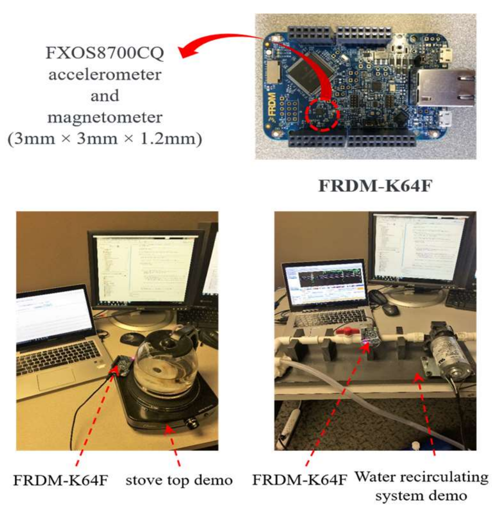



*Fig. 4. IoT applications with FRDM-K64F: intelligent stove top (Appl. A) and water recirculating system (Appl. B).*

# III. IoT Experiments

With regard to IoT applications, we apply our integrated sensor and ML system for an intelligent stove top and water re-circulating system where the embedded sensor system monitors various conditions. For example, the sensors on a stove top can detect whether water in a pot is boiling or simmering, or potentially whether the pot is empty or even contains other liquids. The water re-circulating system diagnoses different conditions of valves so that abnormal operation caused by the motor, valves, or even pipeline defects can be detected. Those conditions are revealed using statistical features extracted from real-time sensor data. We trained the GMM, running locally on the sensor board, to recognize multiple operating states. We used accelerometer measurement whose statistical features can identify different conditions in many applications [18]. Fig. 4 shows the demonstration setup. The accelerometer on the FRDM-K64F collects raw vibration data at 800 samples per second and the embedded board conducts the GMM-EM algorithm. 2-dimensional $N = 300$ feature samples $\{\mathbf{x}_n\}_{n=1}^{N}$, which are zero-crossing rate (feature 1) and standard deviation (feature 2). These are computed in real time (other features can also be used). From the initial value of model order e.g. $M = 7$, it sequentially decreases until $M = 1$, merging the closest Gaussian components. The minimum MDL value is determined and the corresponding parameters in $\hat{\theta}$ are saved.

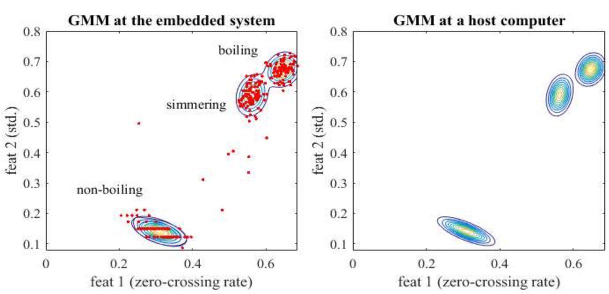
In Fig. 5, we show the comparison of the resulting GMM estimation at FRDM-K64F in real time and the estimates obtained by MATLAB off-line. The trained GMM can be used to cluster (or classify, if labeled features are available) feature samples by assigning them into an appropriate Gaussian component that provides the largest probability. All the training and testing were performed in the embedded board FRDM-K64F and compared with MATLAB based off-line estimates. Plots for the NXP sensor board results were also created by porting data to MATLAB. Table 1 shows the comparison of the average log-likelihood values for the two applications. Once a GMM is trained with $\mathbf{X}_{\text{train}}$, $\hat{\theta}$ is tested with new measurements $\mathbf{X}_{\text{test}}$ from different conditions illustrated in Fig. 5. The testing is performed with 30 samples for each experiment. As a reference of the performance, we also compare our results with the values computed by *gmdistribution* MATLAB toolbox after logging the measurements. Since the EM algorithm is affected by initial parameter values, which are randomly selected, the procedure above is repeated 20 times and the results are averaged. Table 1 shows that the estimated GMM by the embedded system is close to the off-line results obtained by the host computer.

TABLE I. COMPARISON OF GMM ESTIMATION ACCURACY

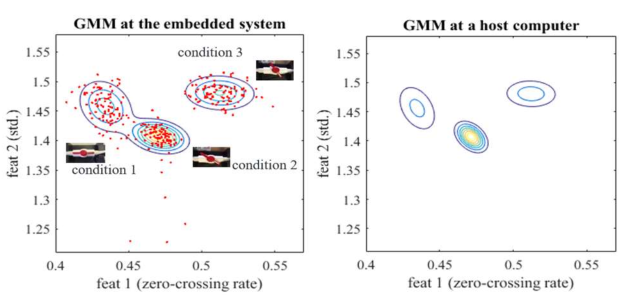
| ***Log-likelihood*** | $X_{train}$ | $X_{test}$ |
|---|---|---|
| Appl. A with $\hat{\theta}_{embedded}$ | 1.9987 | 1.4629 |
| Appl. A with $\hat{\theta}_{off-line}$ | 2.1009 | 1.4686 |
| Appl. B with $\hat{\theta}_{embedded}$ | 2.5877 | 1.9696 |
| Appl. B with $\hat{\theta}_{off-line}$ | 2.7472 | 1.9364 |

## IV. CONCLUSION

We considered an integrated sensor/machine learning system that is suitable for certain IoT applications. The embedded algorithm automatically determines an appropriate set of GMM parameters using the embedded processor/sensor board in real time, and can provide classification results continuously. The embedded implementation is compared to a similar offline MATLAB implementation and the results reveal comparable classification and clustering performance. Our integrated sensor/ML system provided real-time analytics with on-demand training. This system can also be used for other applications including those that involve sensor fusion of data obtained from heterogeneous sensors. The ability of the system to classify signals and provide analytics directly from the board enables lower sensor data traffic.




*(a) Stove top ($d = 2$)*




*(b) Water recirculating system ($d = 2$)*

*Fig. 5. Comparison between the estimated GMMs at the real-time embedded system (left panels) and off-line (right panels).*

## REFERENCES

[1] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006.

[2] C. Aldrich and L. Auret, *Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods*, Advances in Computer Vision and Pattern Recognition. Springer, 2013.

[3] H. Song, J. Thiagarajan, Karthikeyan N. Ramamurthy, A. Spanias, and P. Turaga, "Consensus inference on mobile phone sensors for activity recognition," in *Proc.* of *IEEE ICASSP*, pp. 2294–2298, Mar. 2016.

[4] S. Patel et al, "A review of wearable sensors and systems in rehabilitation.," *Journal Neuroeng. & Rehab.* pp. 1–17, April 2012.

[5] D. Rajan, A. Spanias, S. Ranganath, M. Banavar, and P. Spanias, "Health monitoring laboratories by interfacing physiological sensors to mobile android devices," in *Proc. IEEE FIE*, 2013.

[6] X. Zhang, M.K. Banavar, M. Willerton, A. Manikas, C. Tepedelenlioglu, A. Spanias, T. Thornton, E. Yeatman, and A.G. Constantinides, "Performance comparison of localization techniques for sequential wsn discovery," in *Proc. of IEEE SSPD*, Sept 2012.

[7] X. Zhang, C. Tepedelenlioglu, M. Banavar, and A. Spanias, "Distributed location detection in wireless sensor networks," *Proc. of IEEE Asilomar Conf. on Signals, SystemsComp*, pp. 428–432, Nov 2013.

[8] 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, March 2006.

[9] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[10] H. Hu, Y. Wen, T.-S. Chua, X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*,, pp. 652–687, 2014.

[11] A. Ukil, S. Bandyopadhyay, and A. Pal, "Iot-privacy: To be private or not to be private,"*Proc. ICCC Workshops*, pp. 123–124, April 2014.

[12] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, biggest growth in the far east," Tech. Rep., IDC, 2012.

[13] I. Chen, "Sensor data analytics – unlocking value in 'big data'," [online] *EE Times* Blog - Designlines Internet of Things, 2015, Oct. 10, 2016.

[14] NXP Semiconductor, *Intelligent Sensing Framework v. 2.0 Software Reference Manual for the Kinetis Family of Microcontrollers*, 2015,.

[15] A. Dempster et al, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Society. Series B*, pp. 1–38, 1977.

[16] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Ann. Statist*., vol. 11, no. 2, pp. 416–431, 1983.

[17] C. Bouman, "Cluster: An unsupervised algorithm for modeling Gaussian mixtures," http://engineering.purdue.edu/~bouman, April 1997.

[18] M. Stanley, "Mining sensor data," NXP Blog, [online] available: http://blog.nxp.com/author/michael-stanley, 2015, Oct. 10, 2016.

[19] S. Earley, "Analytics, Machine Learning, and the Internet of Things,"*IT Professional,* vol. 17, issue 1, Jan.-Feb. 2015.

[20] Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020", Dec. 2013, [online] Available: http://www.gartner.com/newsroom/id/2636073, Oct. 10, 2016.

[21] P. Vlacheas et al, "Enabling smart cities through a cognitive management framework for the internet of things," *IEEE Comm. Mag.,* vol. 51, no. 6, June 2013.

[22] M. Naphade, "Smarter Cities and Their Innovation Challenges", *IEEE Computer Society*, vol. 44, no. 6, pp. 32-39, June 2011.

[23] A. Spanias, T. Painter, V. Atti, *Audio Signal Processing and Coding*, Hardcover ISBN: 9780471791478, Wiley, Textbook, March 2007.

[24] D. Schatsky, "Machine learning is going mobile," available: http://dupress.deloitte.com/dup-us-en/focus/signals-for-strategists/machine-learning-mobile-applications.html, April, 2016,.

[25] M. T. Khorshed e al, "Integrating Internet-of-Things with the power of Cloud Computing and the intelligence of Big Data analytics — A three layered approach," in *Proc. of APW C*, December 2015.

[26] T. Painter and A. S. Spanias, "Perceptual Coding of Digital Audio," *Proc. of the IEEE*, pp. 451-513, vol. 88, no. 4, April 2000 .

[27] P. Loizou and A. Spanias, "High Performance Alphabet Recognition," *IEEE Trans. on Speech and Audio,* v. 4, 6, pp. 439-445, Nov. 1996.

[28] L. Deng, "Deep Learning: from Speech Analysis/Recognition to Language/Multimodal Processing," Deep Learning Technology Center, Microsoft Research, Redmond, WA. A Tutorial at ICML, June 21, 2014.

[29] A. Spanias, N. Tao, A. Hurtado, L. Hartwell, B. Greenes, "An integrated traineeship program for sensors and algorithms for health monitoring," Abstract, Poster Session, SensMach 2016, Scottsdale, Nov. 2016.

[30] F. Rejab, K. Nouira, A. Trabelsi, "Health Monitoring Systems Using Machine Learning Techniques," *Intel. Syst. Sc. & Info.,* Springer, 2014.

[31] G. Wishon, "Sensors, IoT, and Smart Campuses: Challenges and Opportunities," Abstract, SensMach 2016, Scottsdale, November 2016.

[32] R. Thomas,"Speech Research at Amazon," Plenary Talk, *IEEE DSP & SPE Workshop*, Snowbird, Utah, August 2015.

[33] B. Mears and M. Shah, *Virtual Design of an Audio Lifelogging System: Tools for IoT Systems*, Ed. A. Spanias, Synth. Lectures on Algorithms and Software in Engineering, vol. 8, no. 1, pp. 1-73, June 2016.

[34] C. Wren et al., "Pfinder: Real-Time Tracking of the Human Body," *IEEE Trans. PAMI*, pp. 780-785, vol 19, no 7, July 1997.

[35] A. S. Spanias, "Speech Coding: A Tutorial Review," *Proceedings of the IEEE*, vol. 82, no. 10, pp. 1441-1582, October 1994,

[36] A. Spanias, *Digital Signal Processing; An Interactive Approach – **2nd Edition**,* 403 pages, ISBN 978-1-4675-9892-7, Lulu Press On-demand Publishers Morrisville, NC, May 2014.

[37] J. J. Verbeek, N. Vlassis, B. Kröse, "Efficient Greedy Learning of Gaussian Mixture Models," *Neural Comp.*, vol. 15, 2, p. 469, Feb. 2003.

[38] V. Berisha, A. Wisler, A. Hero, A. Spanias, "Empirically Estimable Classification Bounds Based on a Nonparametric Divergence Measure," *IEEE Tran. on Signal Processing,*, vol. 64, 3, pp.580-591, Feb. 2016.

[39] S. Theodorides, *Machine Learning, A Bayesian and Optimization Perspective, 1st Edition*, Academic Press, December 2015.

[40] S. Hartshorn, Bayes Theorem Examples: An Intuitive Guide, Jan. 2016.

[41] H. Braun, P. Turaga, A. Spanias, G. Tepedelenlioglu, 'Direct Classification from Compressively Sensed Images via Deep Boltzmann Machine," *Proc. IEEE Asilomar Conf. on SSC*, Monterey, Nov. 2016.

[42] X. Xu, L. Xie, S. Wang. "Multimode process monitoring with PCA mixture model." *Computers & Electrical Eng.* 40:72101-2112, 2014,

[43] 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*, , T1.SP1.12, Limassol, April 2016.

[44] N. Kovvali, M. Banavar, A. Spanias, *An Introduction to Kalman Filtering with MATLAB Examples*, Synthesis Lectures on Signal Processing, Morgan & Claypool Publishers, Ed. Jose Mura, vol. 6, no. 2, pp. 1-81, ISBN 13: 9781627051392, September 2013.

[45] N. Greggio, A. Bernardino, P. Dario, J. Santos-Victor. "Efficient greedy estimation of mixture models through a binary tree search." *Robotics and Autonomous Systems*, 62:101440-1452, 2014.

[46] K.. Ramamurthy and A. Spanias, *MATLAB® Software for the Code Excited Linear Prediction Algorithm: The Federal Standard-1016*, Morgan and Claypool Publishers, vol. 2, no. 1, pp. 1-109: 1-109, ISBN 1608453847, January 2010.

[47] Y. Song, Z. Ji, Q. Sun, "An extension Gaussian mixture model for brain MRI segmentation," in *Proc.* of *IEEE EMBC.* pp. 4711-4714, 2014.

[48] Y. Wang, K. Cao, "A Proactive Complex Event Processing Method for Large-Scale Transportation Internet of Things," *International Journal of Distributed Sensor Networks* 20141-8., January 2014.

[49] J. Makhoul, S. Roucos, H. Gish, "Vector quantization in speech coding," *Proc. of the IEEE,* 73 (11), 1551-1588, 1985.

[50] Y. Linde, A. Buzo, R. M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. On Comm.,* Vol. Com-28, No. 1, January 1980.

[51] J. Guo, H. Zhang, Y. Sun, R. Bie "Square-root unscented Kalman filtering-based localization and tracking in the Internet of Things," *Personal and Ubiquitous Computing*, v. 18, 4, pp 987–996, April 2014.

[52] NXP Application Note AN5018, Rev. 2.0: Basic Kalman Filter Theory.

[53] NXP Application Note AN5023, Rev.2.0: Sensor Fusion Kalman Filters.

[54] S. Peshin, D. Ramirez, J. Lee, H. Braun, C. Tepedelenlioglu, and A. Spanias, "A Photovoltaic (PV) Array Monitoring Simulator," in *Proc. of International Conference on MIC,* 2015.

[55] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, v29, pp. 1645 – 1660, September 2013.

[56] C. C. Aggarwal, *Managing and Mining Sensor Data*, Springer, 2013.

[57] X. Bi, S. Lee, J. F. 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," *Journal of Analytical Atomic Spectrometry*, vol. 29, issue 9, pp. 1630 – 1639, 2014.

[58] G. Wichern, J. Xue, H. Thornburg, B. Mechtley, A. Spanias, "Segmentation, Indexing, and Retrieval for Environmental and Natural Sounds," *IEEE Transactions on ASLP*, vol. 18, pp. 688 -707, 2010.

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

[60] 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," in *Proc. of IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 1779 – 1783, November 2015.

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

[62] P. Sattigeri, J. J. Thiagarajan, K. N. Ramamurthy, A. Spanias, M. Goryll and T. Thornton, "Robust PSD Features for Ion-Channel Signals," *Proc. of the SSPD*, London, UK, 27-29 September 2011.

[63] H. Braun, S. T. Buddha, V. Krishnan, C. Tepedelenlioglu, A. Spanias, M. Banavar, and D. Srinivansan, "Topology reconfiguration for optimization of photovoltaic array output," *Elsevier Sustainable Energy, Grids and Networks* (*SEGAN),* pp. 58-69, Vol. 6, June 2016.

[64] M. Banavar . J. Zhang , B. Chakraborty , H. Kwon, Y. Li, H. Jiang, A. Spanias, , C. Tepedelenlioglu, C. Chakrabarti A. Papandreou-Suppappola,"An overview of recent advances on distributed and agile sensing, algorithms and implementation," *DSP*, Elsevier, 2015.

[65] Valentina Baljak, Kenji Tei, Shinichi Honiden, "Fault classification and model learning from sensory Readings — Framework for fault tolerance in wireless sensor networks", *IEEE 8th Int. Conference on, Intelligent Sensors Sensor Networks and Information Proc.,* pp. 408-413, 2013.

[66] A. Dimitrievski, E. Zdravevski, P. Lameski, V. Trajkovik, "A survey of Ambient Assisted Living systems: Challenges and opportunities," *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP),* pp. 49-53, 2016.

[67] Stefan Bosse. "Mobile Multi-agent Systems for the Internet-of-Things and Clouds Using the JavaScript Agent Machine Platform and Machine Learning as a Service," *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 244 - 253, 2016.

[68] Satyavrat Wagle, "Semantic data extraction over MQTT for IoTcentric wireless sensor networks," *2016 International Conference on Internet of Things and Applications (IOTA),* pp. 227 - 232, 2016.