Extreme Learning Machines for Datasets with Missing Values Using the Unscented Transform

Diego P. P. Mesquita, João P. P. Gomes
Department of Computer Science
Federal University of Ceará
Rua Campus do Pici, sn
Fortaleza, Ceará, Brazil, 60440-554
Email: {diegoparente, jpaulo}@lia.ufc.br

Leonardo R. Rodrigues
Electronics Division
Institute of Aeronautics and Space
Praça Marechal Eduardo Gomes, 50
São José dos Campos, São Paulo, Brazil, 12228-904
Email: leonardolrr@iae.cta.br

***Abstract*—The existence of missing data is a common fact in real applications which can significantly affect the data analysis process. In order to overcome this problem, many methods have been proposed in the literature. Extreme Learning Machine (ELM) has become a very popular research topic in machine learning and artificial intelligence areas due to its characteristics such as fast training procedure, good generalization and universal approximation capability. Although ELM has been successfully applied in different domains, its basic formulation cannot handle datasets with missing values properly. This paper presents a variant of the Extreme Learning Machine (ELM) for datasets with missing values. In the proposed method, probability distributions for the missing values are estimated using the expectation maximization (EM) algorithm, assuming that data is normally distributed. The Unscented Transform (UT) is used to estimate the values of the hidden layer outputs, and the weights of the output layer are assigned using the Moore-Penrose Pseudoinverse. Numerical experiments are carried out in order to evaluate the performance of the proposed method in four real world and two synthetic regression datasets. The results show that the proposed method presented a good performance in terms of Average Root-Mean-Squared Error (ARMSE).**

***Keywords*—Extreme Learning Machines, Unscented Transform, Missing Values.**

I. INTRODUCTION

Missing data is a common fact that can have significant effect in data analysis. The missing data phenomenon can be defined as the absence of some entries in feature vectors caused by factors such as measurement error, device malfunction and operator failure, among others [1]. According to Little and Rubin in [2] the missingness mechanism can be classified as Missing Completely at Random (MCAR), Missing at Random (MAR) and Not Missing at Random (NMAR). When data is MCAR, the probability of an instance having a missing value for an attribute does not depend on either the observed data or the value of missing attribute itself. If data is MAR, the probability of an instance having a missing value for an attribute may depend on the known values, but not on the actual value of the missing entry. In NMAR, the probability of an instance having a missing value for an attribute may depend on the value of that attribute [3]. In this paper, we assume that the missing values are MAR.

Since the pioneering work of Rubin [4], many strategies have been proposed to handle the missing data problem. The existing approaches can be roughly classified in three groups: deletion of incomplete cases, imputation methods and reformulation of data analysis/machine learning methods.

Deletion of incomplete cases is the simplest strategy and probably the most used. In such approach, instances in which any attribute is missing are not considered on the training set. This approach is formally referred as listwise deletion. Although very popular, listwise deletion may severely impact the final result when the number of instances with missing data is significant [5]. The imputation approach consists of filling the missing values using information extracted from the observed values in the same attribute vector and the whole dataset. Imputation methods can be divided into single and multiple imputation. While single imputation methods aim to fill the missing values with a single value, in multiple imputation, multiple versions of the dataset are generated by imputing missing values in different ways and a model is generated for each dataset version. The final model is a combination of the individual models. As can be inferred, single imputation methods are simpler and often require less computational effort. Multiple imputation, on the other hand, is a more computationally demanding process but addresses the inherent uncertainty in the process of estimating a missing value. As a consequence, multiple imputation often presents better results.

The last approach consists in adapting well known methods so that missing data can be handle in its formulation without any explicit imputation. Examples of such approach can be found for Support Vector Machines [6], Minimal Learning Machines [7] and K-nearest neighbors [8], among others. According to [7], this approach is preferred since it handles the uncertainty of the estimation process while keeping the computational cost on the same level as single imputation. It is worth pointing that in most of these methods it is necessary to estimate the distribution of the dataset using any parametric or nonparametric model. The model parameters are estimated using any method that is robust to the presence of missing values. Usual choices include the use of the Expectation-Maximization (EM) algorithm for datasets with missing values to estimate the parameters of Gaussian [9] and Mixture of Gaussian distributions [10].

Extreme Learning Machine (ELM, [11]) is a single layer neural network that has attracted much attention due to its

performance in many tasks and the fast training procedure [12]. In ELM, the weights of the hidden layer are assigned randomly and the output layer weights are estimated using the Moore-Penrose pseudoinverse. Despite its successful application in many domains, the basic formulation of ELM cannot handle datasets with missing values appropriately. Existing approaches include single [13] and multiple imputation [1] algorithms. However, no modified version of ELM to handle missing values is available.

In this paper, we propose a variant of the ELM for datasets with missing values. In the proposed method, probability distributions for the missing values are estimated using the EM algorithm, assuming that data is normally distributed. After that, the expected values of the hidden layer outputs are estimated using the Unscented Transform (UT, [14]). Finally, the output layer weights are assigned using the Moore-Penrose Pseudoinverse. The performance of the proposed method is assessed in experiments with four real world and two synthetic regression datasets.

The remaining sections of this paper are organized as follows. Section II presents a brief overview of ELM and Unscented Transform. Section III introduces the proposed method. Section IV shows the results obtained in the numerical experiments. Concluding remarks are presented in Section V.

II. THEORETICAL BACKGROUND

A. Extreme Learning Machines

In recent years, Extreme Learning Machine (ELM) has become a popular research topic in machine learning and artificial intelligence areas due to its characteristics such as fast training, good generalization and universal approximation capability [15].

ELM was firstly introduced by [11]. It is a learning algorithm for single hidden layer feedforward neural networks (SLFNs) in which the hidden node parameters are randomly generated and the output weights are analytically computed [16]. Many works have been published recently showing the use of ELMs with good performance in a wide range of applications such as image segmentation [17] and face recognition [18], among others.

Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ be the training dataset, such that $\mathcal{X} = \{x_i\}_{i=1}^{N}$ is the set of p-dimensional training inputs and $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ is the set of their respective q-dimensional outputs. Consider a SLFN with $\mathcal{M}$ neurons in the hidden layer and activation function $\phi(\cdot)$. The output of the SLFN for data x_i, denoted by s_i, can be expressed according to (1).

$$s_i = \sum_{i=1}^{\mathcal{M}} \beta_i \phi\left(w_i \cdot x_i + c_i\right) \tag{1}$$

where w_i is the weight vector connecting the input layer and the i-th hidden neuron, β_i is the weight vector connecting the i-th hidden neuron and output layer and c_i is the threshold of the i-th hidden neuron [19].

It is possible to express (1) using a matrix notation $H\beta$, where H is the hidden layer output matrix [19]. Each element h_{ij} of matrix H, with $1 \leq i \leq N$ and $1 \leq j \leq \mathcal{M}$ is defined by (2).

$$h_{ij} = \phi\left(w_j \cdot x_i + c_j\right) \tag{2}$$

Then, the weights of the hidden layer can be obtained by applying the method of least squares in order to find an approximated solution for the linear system $H\beta = Y$, in which each row of Y corresponds to y_i. This approximated solution can be obtained through (3).

$$\hat{\beta} = \left(H^T H\right)^{-1} H^T Y \tag{3}$$

In this paper, we use the sigmoid function presented in (4) as the activation function of the neurons in the hidden layer.

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{4}$$

B. Unscented Transform

The Unscented Transform (UT), originally proposed in [20], is a method for estimating statistical moments of a probability distribution associated to a random variable which results from a nonlinear transformation of another random variable [21].

Let x be a d-dimensional random variable and $\phi(\cdot)$ be an arbitrary nonlinear function. Let $h = \phi(x)$ be the transformed random variable, which is obtained by the transformation of x through $\phi(\cdot)$.

In order to approximate h using the UT method, a set $\mathcal{S} = \{\gamma_l\}_{l=1}^{L}$ of sigma points (SPs), with respective weights $\{k_l\}_{l=1}^{L}$, associated to the original random variable x are deterministically chosen. Then, the SPs are transformed through $\phi(\cdot)$, resulting in a transformed set of SPs. Finally, the transformed SPs (and their corresponding weights) are used in order to approximate the statistical moments of h.

The weights k_l, with $1 \leq l \leq L$, can be positive or negative. However, they must obey the condition in (5) in order to provide an unbiased estimate [14].

$$\sum_{l=1}^{L} k_l = 1 \tag{5}$$

The implementation of the UT is based on the expressions presented in equations (6) to (8).

$$\delta_l = \phi\left(\gamma_l\right) \qquad \forall\, 1 \leq l \leq L \tag{6}$$

$$\bar{h} = \sum_{l=1}^{L} k_l \delta_l \tag{7}$$

$$\Sigma_h = \sum_{l=1}^{L} k_l \left(\delta_l - \bar{h}\right)\left(\delta_i - \bar{h}\right)^T \tag{8}$$

where $\bar{h}$ and Σ_h are the estimated mean and covariance matrix of h, respectively; γ_l and δ_l are the SPs associated to x and h,

respectively; k_l is the weight of γ_l. Note that a SP may have different weights in the approximation of $\bar{h}$ and Σ_h.

There are different ways to choose the SPs. A common approach is to use a symmetric set of $L = 2d+1$ SPs, chosen according to Equations (9) to (12).

$$\gamma_1 = \bar{x} \tag{9}$$

$$\gamma_l = \bar{x} + \left[\sqrt{d\,\Sigma_h}\right]_{l-1} \qquad \forall\, 1 < l \leq d+1 \tag{10}$$

$$\gamma_l = \bar{x} - \left[\sqrt{d\,\Sigma_h}\right]_{l-(d+1)} \qquad \forall\, d+1 < l \leq 2d+1 \tag{11}$$

$$k_l = \frac{1}{2d+1} \qquad \forall\, 1 \leq l \leq 2d+1 \tag{12}$$

where $\left[\sqrt{d\,\Sigma_h}\right]_l$ denotes the i-th row of the matrix square root of $d\,\Sigma_h$, which is the original covariance matrix Σ_h multiplied by the number of dimensions d.

III. PROPOSED METHOD

In order to proceed with the computation of (3) in the presence of missing data, we propose taking the expected value of each entry of H. If $x_i \in \mathcal{X}$ is fully observed, each h_{ij} can be directly computed using (2). We are left with the more interesting case, in which x_i is missing one or more entries. Let M denote the indices of x_i which contain missing values and O denote the observed ones. Suppose each element in $\mathcal{X}$ was drawn from a same multivariate normal distribution with mean vector μ and covariance matrix Σ. It follows from this assumption that:

$$x_{iM} \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{13}$$

in which μ_* and Σ_* are the conditional mean vector and covariance matrix with respect to the observed entries x_{iO} of x_i, given by (14) and (15), respectively.

$$\mu_* = \mu_M + \Sigma_{MO}\Sigma_{OO}^{-1}(x_{iO} - \mu_O) \tag{14}$$

$$\Sigma_* = \Sigma_{MM} - \Sigma_{MO}\Sigma_{OO}^{-1}\Sigma_{OM} \tag{15}$$

Therefore, h_{ij} is a nonlinear transform of x_{iM}, for which we wish to compute its expectation $\mathrm{E}[h_{ij}]$, given by

$$\mathrm{E}[h_{ij}] = \int_{\mathbb{R}^{|M|}} \phi(w_{jM} \cdot x + \zeta)\mathcal{N}(x|\mu_*, \Sigma_*)dx \tag{16}$$

where $\zeta = w_{jO} \cdot x_{iO}$ and $\mathcal{N}(x|\mu_*, \Sigma_*)$ denotes the value of the multivariate normal *pdf* defined by the mean vector μ_* and covariance matrix Σ_* evaluated at x. Since ϕ is the sigmoid function, eq. (16) resumes to

$$\mathrm{E}[h_{ij}] = \int_{\mathbb{R}^{|M|}} \frac{1}{1 + e^{-(w_{jM} \cdot x + \zeta)}}\mathcal{N}(x|\mu_*, \Sigma_*)dx \tag{17}$$

which is intractable. The problem of estimating the distribution (or its statistical moments) of a random variable resulting from a nonlinear transformation of another random variable can be addressed by either a Monte Carlo approach or some approximations. In a Monte Carlo estimation procedure, samples of the original distribution would be transformed by the nonlinear function and the transformed variables could be used to estimate the distribution itself or its moments. As a consequence of the sampling procedure, this approach results on an increase in the computational cost that could even reach a cost similar to multiple imputation.

Considering the approximations, two strategies are most commonly used: analytical and statistical linearization. Analytical linearization methods may use expansions like Taylor expansion to approximate the nonlinear function as a set of linear components. On the other hand, statistical linearization procedures aim to approximate several moments of the distribution using the nonlinear transform and a set of previously defined sampled sigma points [22]. It has been reported that statistical linearization methods outperform analytical linearization methods [23]. In this paper, the Unscented Transform is used as the statistical linearization method.

Once the sigma points have been sampled, the expected value of the variables of interest (its first noncentral moment) can be approximated by:

$$\mathrm{E}[h_{ij}] \approx \sum_{l=1}^{L} k_l \phi(w_j \cdot \gamma_l + \zeta) \tag{18}$$

The proposed method is outlined in Algorithm 1 and is hereinafter referred to as U-ELM.

Algorithm 1 Unscented ELM

procedure U-ELM($\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$)
  Estimate μ, Σ from $\mathcal{X}$ using Expectation Maximization
  for $i = 1 \ldots N$ **do**
    Let M be the set of unknown indices in x_i
    Let O be the set of observed indices in x_i
    for $j = 1 \ldots \mathcal{M}$ **do**
      if $M \neq \emptyset$ **then**
        $\mu_* \leftarrow \mu_M + \Sigma_{MO}\Sigma_{OO}^{-1}(x_{iO} - \mu_O)$
        $\Sigma_* \leftarrow \Sigma_{MM} - \Sigma_{MO}\Sigma_{OO}^{-1}\Sigma_{OM}$
        $\zeta \leftarrow w_{jO} \cdot x_{iO}$
        $L \leftarrow 2|M| + 1$
        $k_{1\ldots L} \leftarrow L^{-1}$
        $\gamma_1 \leftarrow \mu_*$
        for $l = 2 \ldots (|M| + 1)$ **do**
          $\gamma_l \leftarrow \gamma_1 + \left[\sqrt{|M|\,\Sigma_*}\right]_{l-1}$
        end for
        for $l = (|M| + 2) \ldots L$ **do**
          $\gamma_l \leftarrow \gamma_1 - \left[\sqrt{|M|\,\Sigma_*}\right]_{l-(|M|+1)}$
        end for
        $h_{ij} \leftarrow \sum_{l=1}^{L} k_l \phi(w_j \cdot \gamma_l + \zeta)$
      else
        $h_{ij} \leftarrow \phi(w_j \cdot x_i + c_j)$
      end if
    end for
  end for
  $\hat{\beta} \leftarrow \left(H^T H\right)^{-1} H^T Y$
end procedure

IV. NUMERICAL EXPERIMENTS

The most common techniques to deal with missing data in regression problems consist on either dropping the examples counting on missing entries or to directly impute the conditional mean of the missing entries (CMI), which can be estimated using EM. To evaluate the performance of the proposed method, we compare it with conventional ELMs trained on datasets treated with these techniques.

To carry out the experiments, four real world and two synthetic regression datasets were used. The real-world datasets were all obtained from the UCI machine learning repository [24]. The synthetic datasets were generated from the 3D Sinc Function and the Smoothed Parity Function, given by (19) and (20) and illustrated in Figures 1 and 2, respectively. Further details about each dataset are presented in table I.

$$f_{Sinc}(x,y)=\frac{sin(\sqrt{x^2+y^2})}{\sqrt{x^2+y^2}} \tag{19}$$

$$f_{xor}(x,y) = sin(2\pi x)sin(2\pi y) \tag{20}$$

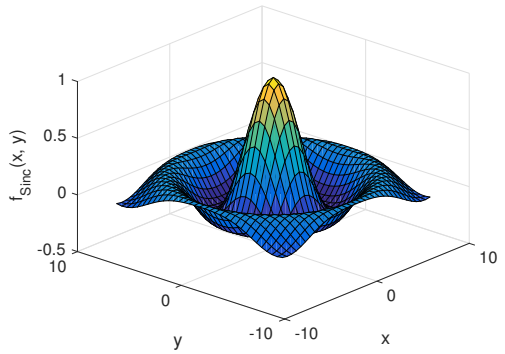


Fig. 1: 3D Sinc Function

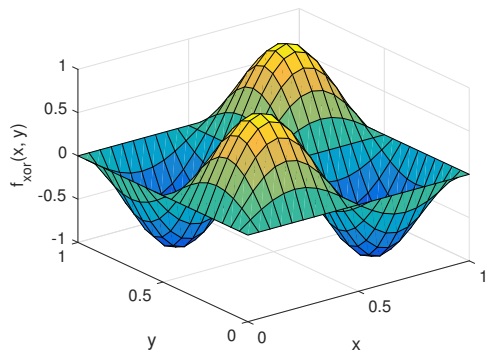


Fig. 2: Smoothed Parity Function

TABLE I: Datasets characteristics

Dataset	#attributes	#Train	#Test
Bank	8	2999	1500
Boston Stocks	9	633	317
Boston Housing	13	337	169
Concrete Compression	8	686	344
Smoothed Parity Function	2	294	127
3D Sinc Function	2	704	352

To access the impact of the number of missing values in the proposed method, we gradually increase the number of examples with missing attributes in each dataset. Experiments consisted on 10 similar rounds of experiments. In each of these rounds, the training and test samples were drawn randomly without replacement from the original data sets. The artificial missing values are generated in a manner to guarantee that data is MAR. Each ELM was trained using $\mathcal{M} = 100$ neurons in its hidden layers. All implementations were executed using MATLAB. The results are presented in terms of Average Root-Mean-Squared Error (ARMSE) in Figure 3.

As expected, the listwise deletion has the worst performance in most datasets. This fact is more noticeable when the number of missing values is increased. The U-ELM had the best overall performance, achieving the lowest ARMSE values in all datasets for all rates of training examples containing missing values.

The performance gap between CMI and U-ELM can be explained by observing the formulation of each regression model. Using the CMI results in an ELM model that can be described by:

$$s_i = \sum_{i=1}^{\mathcal{M}} \beta_i \phi \left(w_i \cdot E[x_i] + c_i \right) \tag{21}$$

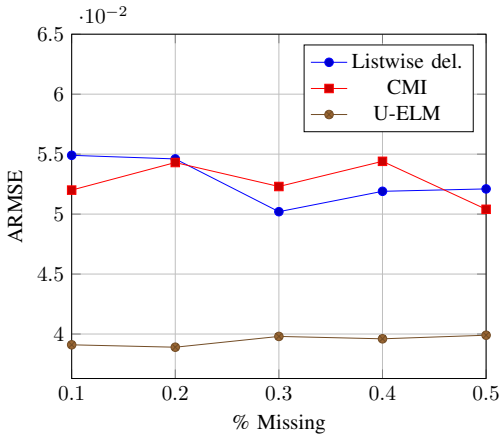
where $E[x_i]$ is a vector composed by the observed values of x_i and the expected values of the missing attributes conditioned to the observed ones estimated with the CMI.

It can be noticed that, in the CMI, the ELM (with its nonlinear transforms) is applied to a dataset where the missing values are filled with its expected values. On the other hand, the U-ELM estimates the expected value of the whole ELM network, not only the input vectors x_i. Considering (1), (2) and (18), it is possible to describe the U-ELM as:

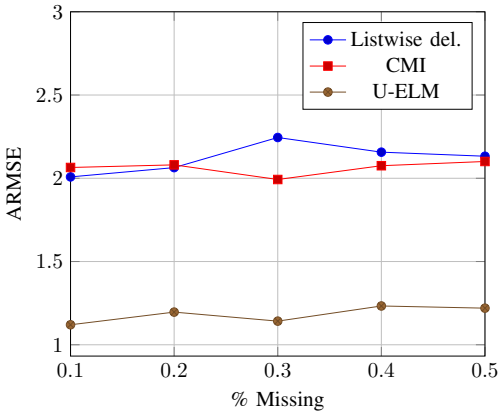
$$s_i = \sum_{i=1}^{\mathcal{M}} \beta_i E[\phi \left(w_i \cdot x_i + c_i \right)] \tag{22}$$

which leads to:

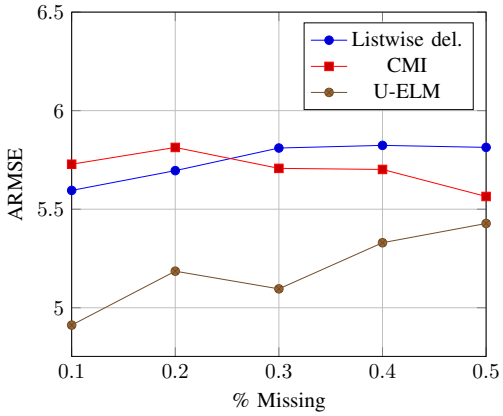
$$s_i = E\left[\sum_{i=1}^{\mathcal{M}} \beta_i \phi \left(w_i \cdot x_i + c_i \right) \right] \tag{23}$$



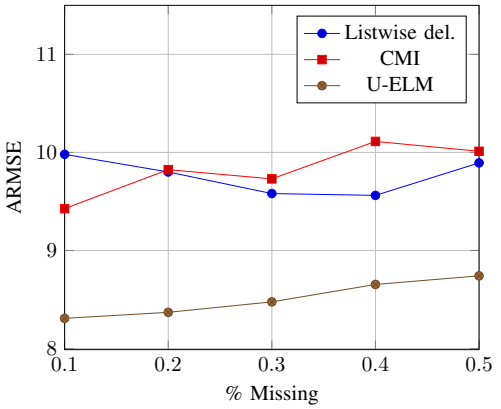
(a) Bank



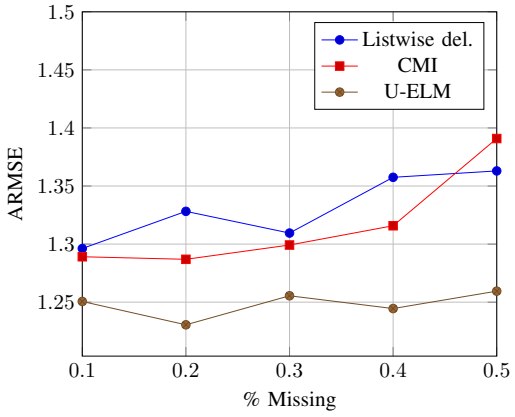
(b) Boston Stocks



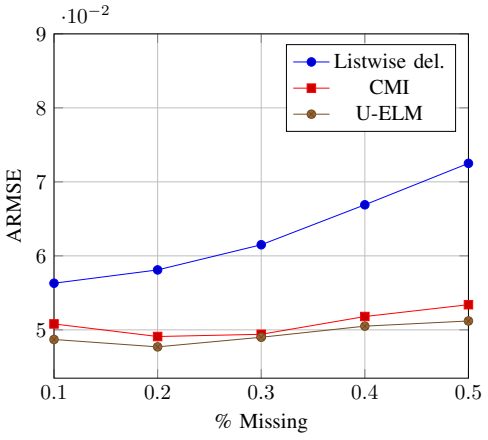
(c) Boston Housing



(d) Concrete Compression



(e) Smoothed Parity Function



(f) 3D Sinc Function

Fig. 3: Comparison between U-ELM and standard methods to handle data with missing entries.

V. CONCLUSIONS

This paper presented an extension of ELMs to deal with datasets containing missing data. The proposed approach uses the UT to estimate individually the entries of the transformed input matrix H. It is worth noting that the same approach can be used in test time in order to provide outputs in face of incomplete feature vectors.

Results showed the proposed ELMs using UT achieved the best overall results when compared to the standard strategies to deal with missing data.

Future works may include extensions of U-ELM to make it robust to the uncertainty around h_{ij} and the generalization of its formulation in order to use mixture of gaussians.

ACKNOWLEDGMENTS

The authors acknowledge the support of CNPq (Grant 402000/2013-7 and Grant 456837/2014-0).

REFERENCES

[1] D. Sovilj, E. Eirola, Y. Miche, K.-M. Bjrk, R. Nian, A. Akusok, and A. Lendasse, "Extreme learning machine for missing data using multiple imputations," *Neurocomputing*, vol. 174, Part A, pp. 220–231, 2016.

[2] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. Wiley-Interscience, 2002.

[3] P. Kang, "Locally linear reconstruction based missing value imputation for supervised learning," *Neurocomputing*, vol. 118, pp. 65–78, 2013.

[4] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, pp. 581–590, 1976.

[5] P. J. Garcia-Laencina, J.-L. Sancho-Gomez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: A review," *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, Mar. 2010.

[6] K. Pelckmans, J. D. Brabanter, J. Suykens, and B. D. Moor, "Handling missing values in support vector machine classifiers," *Neural Networks*, vol. 18, no. 56, pp. 684 – 692, 2005, {IJCNN} 2005.

[7] D. P. P. Mesquita, J. P. P. Gomes, and A. H. S. Jr., *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I*, 2015, ch. A Minimal Learning Machine for Datasets with Missing Values, pp. 565–572.

[8] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131, pp. 32 – 42, 2014.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[10] L. Hunt and M. Jorgensen, "Mixture model clustering for mixed data with missing information," *Computational Statistics Data Analysis*, vol. 41, no. 34, pp. 429 – 440, 2003, recent Developments in Mixture Model.

[11] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1, pp. 489–501, 2006.

[12] D. P. P. Mesquita, J. P. P. Gomes, L. R. Rodrigues, and R. K. H. Galvão, "Pruning extreme learning machines using the successive projections algorithm," *IEEE Latin America Transactions*, vol. 13, no. 12, pp. 3974–3979, Dec 2015.

[13] Q. Yu, Y. Miche, E. Eirola, M. van Heeswijk, E. Sverin, and A. Lendasse, "Regularized extreme learning machine for regression with missing data," *Neurocomputing*, vol. 102, pp. 45 – 51, 2013.

[14] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.

[15] A. S. C. Alencar, W. L. Caldas, J. P. P. Gomes, A. H. D. Souza, P. A. C. Aguilar, C. Rodrigues, W. Franco, M. F. D. Castro, and R. M. C. Andrade, "MLM-rank: A ranking algorithm based on the minimal learning machine," in *2015 Brazilian Conference on Intelligent Systems (BRACIS)*, Nov 2015, pp. 305–309.

[16] G. Huang, G. B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Networks*, vol. 61, pp. 32–48, 2015.

[17] C. Pan, D. S. Park, Y. Yang, and H. M. Yoo, "Leukocyte image segmentation by visual attention and extreme learning machine," *Neural Computing and Applications*, vol. 21, no. 6, pp. 1217–1227, 2012.

[18] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2588–2597, 2011.

[19] G. B. Huang and H. A. Babri, "General approximation theorem on feed forward networks," in *Proceedings of the International Conference on Information, Communications and Signal Processing*. IEEE, September 1997, pp. 698–702.

[20] S. J. Julier and J. K. Uhlmann, "Extreme learning machine: theory and applications," in *SPIE Aerosense Symposium*, 1997, pp. 182–193.

[21] B. P. Leão and T. Yoneyama, "On the use of the unscented transform for failure prognostics," in *IEEE Aerospace Conference*. Big Sky: IEEE, March 2011.

[22] M. Farrokhsiar and H. Najjaran, *Autonomous and Intelligent Systems: Second International Conference, AIS 2011, Burnaby, BC, Canada, June 22-24, 2011. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Analysis of Future Measurement Incorporation into Unscented Predictive Motion Planning, pp. 112–123.

[23] R. V. der Merwe and E. A. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 6, 2001, pp. 3461–3464 vol.6.

[24] M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml