IJSA (2022), Vol. 6, No. 1, 1-12
DOI: https://doi.org/10.29244/ijsa.v6i1p1-12

Indonesian Journal of Statistics and Its Applications
Available at https://ijsa.stats.id
e-ISSN: 2599-0802

Implementation of Ensemble Self-Organizing Maps for Missing Values Imputation

Titin Siswantining[1], Kathan Gerry Vivaldi[2], Devvi Sarwinda[3], Saskya Mary Soemartojo[4], Ika Marta Sari[5], Herley Shaori Al-Ash[6]

Department of Mathematics Universitas Indonesia, Depok, Indonesia
‡corresponding author: titin@sci.ui.ac.id

Abstract

The purpose of this study is to implement the ensemble self-organizing maps (E-SOM) method to impute missing values at the preprocessing data stage, which is an important stage when making predictions or classifications. The Ensemble Self-Organizing Maps (E-SOM) is the development of the SOM imputation method, in which the E-SOM method is implemented by applying an ensemble framework using several SOMs to improve generalization capabilities. In this study, the E-SOM imputation method is implemented in South African heart disease data using random forest as a classification model. The results of the model evaluation showed that for accuracy in testing data, the Random Forest model formed from E-SOM imputed data yields better accuracy values than the Random Forest model formed from SOM-imputed data for variations of 36, 49, 64, and 81 neurons, while for variation of 25 neurons both models produce the same accuracy value. From the variation of the number of ensembles applied, the E-SOM imputation method with a combination of 81 neurons and 15 ensemble numbers produced a Random Forest model with the most optimal value of accuracy.

Keywords: ensemble self-organizing maps, imputation, missing values, self-organizing maps

* Received: Jan 2021; Reviewed: Jan 2021; Published: Mei 2022

1. Introduction

Prediction or classification is one of the processes that often carried out in real world data analysis. In the prediction or classification process, proper quality data analysis is needed. Data preprocessing stage is an important stage when analyzing data. One of the problems commonly found when preprocessing data is missing values, which is a condition where a value is not available on certain variables in the data. The condition of missing values can be caused by several factors, such as manual errors at the time of data entry, values that are not available, errors in data retriever equipment, incorrect measurement, and much more (Yadav & Roychoudhury, 2018). Missing values can cause some adverse effects when analyzing data, such as reducing statistical power, creating bias in parameter estimation, reducing sample representativeness, and complicate data analysis (Kang, 2013). Most of the data analysis methods require complete data conditions.

Several ways are used to overcome the missing values, some of which are the case deletion method and the imputation method (Crambes & Henchiri, 2019; Nakagawa, 2015). In the case, deletion observation method containing missing values is omitted, so that the data used is only data that does not include missing values. This method can be used if the percentage of missing values in the data is small. For data with a large portion of missing values, and in the case of small amounts of data, the elimination of observations containing missing values can cause loss of information and can affect the results of the analysis, where the results of the study become less than optimal. The imputation method replaces missing values with appropriate values, such as the mean or value obtained through a particular process or approach, such as regression, neural network, etc. (Folguera et al., 2015a; Nishanth & Ravi, 2016).

Self-Organizing Maps (SOM) is one of the clustering methods with the concept of neural network based on unsupervised learning which is used to represent multidimensional data to lower dimensional space (one or two dimensions) (Folguera et al., 2015a). The SOM imputation method has been used several times in data analysis, both as a clustering method (Bustamam et al., 2018; Köhler et al., 2010) and as an imputation method (Folguera et al., 2015a; Rustum & Adeloye, 2007). In its use as an imputation method, SOM estimates the missing values with the weight of the Best Matching Unit (BMU) component which corresponds to the elements of the input vector containing missing values (Cottrell & Letrémy, 2007). SOM is a promising tool for improving the accuracy of estimations of missing values because its flexibility to be fitted to nonlinear data (Saitoh, 2016).

In 2016, Fumiaki developed the SOM imputation method by implementing an ensemble learning framework called the Ensemble Self-Organizing Maps (E-SOM). Learning Ensemble is an algorithm where several models are combined to improve the generalization ability of the model (Saitoh, 2016). In the E-SOM method several SOMs are applied to the data. Next, the results of the generalizations of each SOM are combined to fill the missing values in the data. In his study, Fumiaki used E-SOM to impute artificial missing values in the complete data and evaluate the results of imputation using the Mean Absolute Error (MAE) measure.

In this paper the SOM and E-SOM methods were applied to impute the missing values in the south African heart disease data. Then a classification model is formed from SOM and E-SOM imputation data using the Random Forest model, and then an evaluation of the classification model that has been formed is carried out.

2. Research Methods

2.1 Self-Organizing Maps (SOM) Algorithm

Self-Organizing Maps (SOM) is an deupervised learning based neural network method commonly used as a clustering method. SOM consists of an input layer consisting of input vectors and output layers that contain neurons that are interconnected with input vectors by weight vectors. The SOM algorithm classifies data by studying patterns or characteristics of observations in the data.

There are three major processes in the SOM algorithm, namely competition: each neuron competes to represent a pattern of input vectors, cooperation: the winning neuron determine the spatial location of excited neurological topological environments, and adaptation: the winning neurons along with the neurons adjacent to it are updated so that they are closer to the input vectors presented (Kubat, 1999). Let i, j, l, and p be a natural number and let $\mathbf{w}_j = [w_{j1}, w_{j2}, w_{j3}, \dots, w_{jn}]$be the weight vector of all nodes, where j is the index of neuron ($j = 1,2,3, \dots, l$), l is the number of neurons, $\mathbf{x}_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ represent the input vector (observation) from the dataset ($i = 1,2,3, \dots, p$), p is the number of the input vector (observation), and n represent the dimension of the input vector. The SOM algorithm is as follows:

Step 1: The number of neuron (l) is specified and the weight vectors of all l neuron are initialized (each vector components contain random values between 0 and 1).

Step 2: Initial learning rate (η_0), initial neighborhood width (σ_0), and the maximum number of the iteration are specified.

Step 3: An input vector is chosen at random from the dataset.

Step 4: The distance between the input vector and all neurons is calculated. The winning neuron/ Best Matching Unit (BMU) is neuron that has the smallest distance to the input vector presented.

$$BMU\ \mathbf{x}_{i(t+1)} = \arg\min_j \left\| \mathbf{x}_{i(t+1)} - \mathbf{w}_j(t) \right\|, t = 0,1,2, \dots \quad (1)$$

Step 5: The BMU and its neighbors are updated

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta(t) h_{BMU(\mathbf{x}),j}(t) \left(\mathbf{x}(t) - \mathbf{w}_j(t) \right) \quad (2)$$

Neighborhood function centered on BMU is as follows

$$h_{BMU(\mathbf{x}),j}(t) = \exp\left(-\frac{d(\mathbf{r}_{BMU}, \mathbf{r}_j)^2}{2\sigma^2(t)} \right), \ t = 0,1,2 \quad (3)$$

Step 6: Repeat Step 2 – Step 5 until the maximum number of the iteration is reached. Every iteration, learning rate, and neighborhood width is updated with the following formulas

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\lambda_1} \right), \ t = 0,1,2, \dots \quad (4)$$

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\lambda_2} \right), \ t = 0,1,2, \dots \quad (5)$$

2.2 Self-Organizing Maps Imputation

The process of imputation using Self-Organizing Maps (SOM) begins by dividing the dataset into two sub-data, namely data containing complete observations, and data containing observations with missing values. After the dataset is divided, the SOM algorithm is applied to the sub-data that contains complete observations until the SOM output is obtained (Cottrell & Letrémy, 2007).

Furthermore, from the sub-data containing observations with the missing values, observations were presented in the SOM produced by the learning process in the sub-data containing complete observations. For each observation presented, the SOM's Best Matching Unit (BMU) is determined. The BMU of the input vectors (observation) presented is determined using the Euclidean distance measure by excluding the missing component of the input vector (observation) from the calculation (Cottrell & Letrémy, 2007; Folguera et al., 2015). BMU is defined as a neuron that has the smallest Euclidean distance with the observations presented using formula is as follows

$$BMU(\mathbf{x}) = \arg\min_{j} \left\| \mathbf{x}_{\{v\backslash k\}} - \mathbf{w}_{\{v\backslash k\}j} \right\| \tag{6}$$

Where $\mathbf{x}_{\{v\backslash k\}}\}$ represents non-missing components of the input vector and $\mathbf{w}_{\{v\backslash k\}j}$ represents the corresponding component of the weight vector.

`After obtaining the weight vector with the smallest Euclidean distance to the observations presented (weight vector of the BMU), an imputation process is carried out by filling in the value of the missing component of the input vector (observation) with the corresponding component of the weight vector of BMU. The Process was carried out on all observations containing missing values.

2.3 Ensemble Self-Organizing Maps Imputation

Ensemble Imputation Method Self-Organizing Maps (E-SOM) is the development of the Self-Organizing Maps (SOM) method. In the SOM method, the more number of neurons used, the quality of the data approximation will also increase, but too many neurons can cause the risk of overfitting conditions in SOM, which will result in reduction of generalization capabilities of SOM. In the E-SOM method, an ensemble framework is applied, a framework that combines several weak learners to improve the learner's generalization ability (Zhou, 2012). This ensemble framework can increase the generalization ability of SOM and reduce the risk of overfitting, so that with increasing generalization ability, the quality of missing values imputations is also expected to increase. Based on the basic concept of ensemble learning, the diversity in each learning algorithm is applied, so that the overall generalization of the model is obtained. The SOM algorithm has a dependency property on the initial value, in this case is the initialization value of the weight vector of each neuron. The learning process results from the SOM algorithm change when the initialization values of the

weight vector change. These properties are used to give diversity to each learner, i.e. each SOM. The E-SOM algorithm is as follows:

Step 1 : The data set is divided into two sub-data, namely sub-data containing complete observations $(\mathrm{x}^{\mathrm{c}})$ and sub-data with observations containing missing values ($\boldsymbol{x}^{m}$).

Step 2 : Weight vectors in each SOM, $\mathbf{w}_{jk}$are initialized with random values between 0 and 1, where $\mathbf{w}_{jk}$is the weight vector for j-th neuron ($j = 1,2,3,\dots,l$) in k-th SOM ($k = 1,2,3,\dots,E$).

Step 3 : E SOM is applied to sub data x^{c} .

Step 4 : By equation (6), the BMU for the k-th SOM is determined for the i-th observation in sub data $\boldsymbol{x}^{m}$.

Step 5 : The final BMU weight vector for i-th observation in sub data $\boldsymbol{x}^{m}$ is obtained by the following formula.

$$\widehat{\mathbf{w}}_{i}^{BMU} = \frac{1}{E}\sum_{k=1}^{E} \mathbf{w}_{\boldsymbol{BMU}_{\boldsymbol{ik}}} \tag{7}$$

Where $\widehat{\mathbf{w}}_{i}^{BMU}$is the final BMU weight vector as a result of the ensemble framework for i-th observation and $\mathbf{w}_{\boldsymbol{BMU}_{\boldsymbol{ik}}}$is the BMU weight vector for i-th oservation in k-th SOM.

Step 6 : The value of the component of the final BMU weight vector is used to fill the corresponding missing component in the i-th observation in sub data $\boldsymbol{x}^{m}$.

Step 7 : Repeat Step 4 – Step 6 until no more observations contain missing values in $\boldsymbol{x}^{m}$.

2.4 Random Forest

Random forest is a machine learning algorithm development of the decision tree algorithm (Maimon & Rokach, 2014), where on the random forest algorithm an ensemble method is applied. The approach of the ensemble method used in the random forest algorithm is bagging (Breiman, 2001). In its work process, random forest applies a bootstrap sampling that produces several data sets, then a decision tree is formed from each of the data sets. In the formation of each tree a random feature selection process is carried out, namely the selection of predictor variables used in tree formation. Each tree produces output, where for regression tree in the form of prediction and classification tree for predictive class voting.

For regression trees, the final output of the Random Forest is determined by the average predicted number of each tree. For classification tree, majority vote (output class with the most frequency) is the final output of Random Forest. Figure 1 Show the illustration of a random forest.

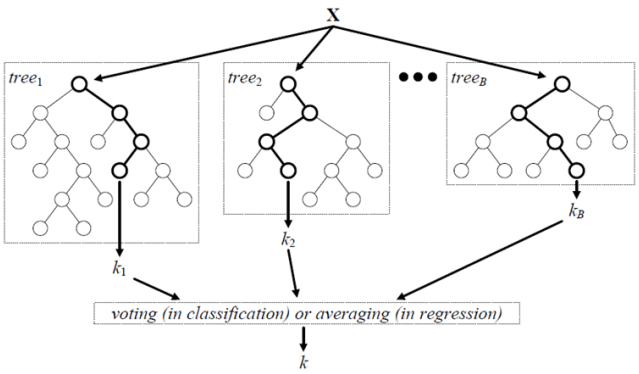

Figure 1: Random Forest Illustration (Verikas et al., 2016)

3. Results and Discussion

3.1 Data

The experiment were performed using South African Heart Disease dataset, which can be accessed from stanford.edu (Friedman et al., 2009) This dataset contains 462 observations with eight continuous variables and two categorical variables. Dataset description is defined in Table 1.

Table 1: Dataset Description

Variables	Definition	Variable Type
sbp	systolic blood pressure	continuous
tobacco	cumulative tobacco (kg)	continuous
ldl	low density lipoprotein cholesterol	continuous
adiposity	adiposity rate	continuous
famhist	family history of heart disease	categorical (present, absent)
typea	type-A behavior	continuous
obesity	obesity rate	continuous
alcohol	current alcohol consumption	continuous
age	age at onset	continuous
chd	response, coronary heart disease	categorical (1=positive chd, 0=negative chd)

3.2 Experimental Setup

The experiments were performed as follows. The dataset was first randomly split into training data and testing data with a ratio of 75:25 so that there will be 324

observations on training data and 138 observations on testing data. After splitting the data, missing values were randomly introduced in the training data. In this study, 5% missing values were used so that there were 102 observations containing missing values (31.48% of a total of 324 observations). Next, the missing values in the training data were imputed using SOM and E-SOM, and then the classification model was built using random forest model on the imputed data.

The model that has been built is then applied to the testing data, so that the accuracy value of the model was obtained. The accuracy value was obtained from calculations using the confusion matrix (Powers, 2020). Figure 2 shows a schematic diagram of the experiments.

3.3 Imputation Stage

Before the imputation process, min-max normalization to the range 0-1 was performed on each variable in the data so that no variable dominates the calculation of Euclidean distance in the mapping process. During the imputation process, categorical variables, namely “famhist” and “chd” are not used in the SOM algorithm.

Table 2:Variations in The Number of Neurons and Neighborhood Width

Number of Neurons	Neighborhood Width
25 (5×5)	2
36 (6×6)	2.5
49 (7×7)	3
64 (8×8)	3.5
81 (9×9)	4

Table 3:Parameter Settings

Parameter	Value
Initial Learning rate (η_0)	0.1
Number of Iterations	1000
Number of Ensembles (E-SOM)	2, 5, 10, 15

The SOM algorithm that was applied to the SOM and E-SOM imputation methods was run using the same conditions and using the same data settings. In both methods, several variations of the number of the number of neurons were applied, and for each variation, different values of neighborhood width were used, as shown in Table 3. The initial neighborhood width was set equal to the “radius” of the output network (KUBAT, 1999).

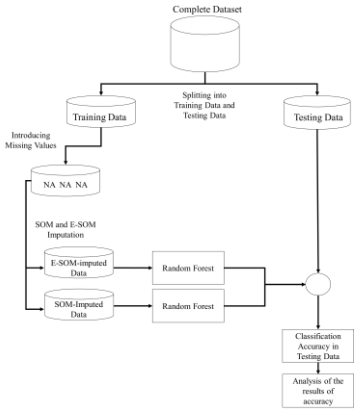

Figure 2: Schematic Diagram of the Experiments.

Parameter settings for each method are shown in Table 3. Each model uses 1000 iterations and the initial learning rate is equal to 0.1. In this study, variations in the number of ensembles were also applied. 5 complete training data for the SOM method and 20 complete training data for the E-SOM method or a total of 25 complete training data will be produced.

3.4 Classification Model Building Stage

After the imputation process, a random forest model was built on each imputed data, then a performance comparison of the models built from SOM-imputed data and E-SOM-imputed data was carried out using accuracy on testing data. First, the “famhist” variable and the “chd” variable were entered again in the data. Then labeling the “famhist” variable was done, where for the "Present" class is labeled "1" and for the "Absent" class is labeled "0". In the process of building the random forest model, hyperparameter tuning was done. The selection of parameters for the building of a model can improve the performance of the model. In this study, hyperparameter tuning was performed on the "number of trees" and "impurity criterion" parameter. The variations applied to the two hyperparameters used are as follows:

1. Number of trees: 80, 100, 120
2. Impurity Criterion: Gini, Entropy

From the two parameters above, 6 models were formed with parameters in the form of a combination of the two hyperparameters. The selection of the best parameter combinations that will be used to build the model was done using the K-Fold Cross Validation method (Friedman et al., 2009, Gareth et al, 2013), with k = 5, where the evaluation of the model for each combination of parameters is the average yield of 5 trials. The final model chosen was the model with the best combination of parameters that produces the highest accuracy rate among the 6 models formed. The final model that was chosen was then used to classify testing data, so that the accuracy value for each model was obtained.

3.5 Experimental Results

Table 2 shows the best models build from SOM-imputed data and E-SOM-imputed data for each number of neurons variations, and Table 3 shows the accuracy values on testing data for each model, where E-SOM uses 15 ensembles.

Table 4:Best Model for Each Method

Number of neurons	Best Model	
	SOM	E-SOM
25	Impurity Criterion: Entropy Number of Trees: 120	Impurity Criterion: Gini Number of Trees: 100
36	Impurity Criterion: Gini Number of Trees: 80	Impurity Criterion: Gini Number of Trees: 100
49	Impurity Criterion: Gini Number of Trees:100	Impurity Criterion: Gini Number of Trees: 100
64	Impurity Criterion: Entropy Number of Trees: 100	Impurity Criterion: Gini Number of Trees: 100
81	Impurity Criterion: Gini Number of Trees: 100	Impurity Criterion: Gini Number of Trees: 120

Table 5:Testing Data Classification Accuracy

Number of neurons	Classification Accuracy	
	SOM	E-SOM
25	0.703	0.703
36	0.696	0.717
49	0.688	0.717
64	0.688	0.717
81	0.703	0.754

Figure 2 shows the comparison chart of the testing data classification accuracy for the model built from SOM-imputed data and E-SOM-imputed data. From Figure 2 it can be seen that for variations of 36, 49, 64, and 81 neurons, random forest model built from SOM-imputed data results in better accuracy values than the model built from E-SOM-imputed data. The random forest model built from imputed data from both methods produces the same accuracy value for 25 neuron variations.

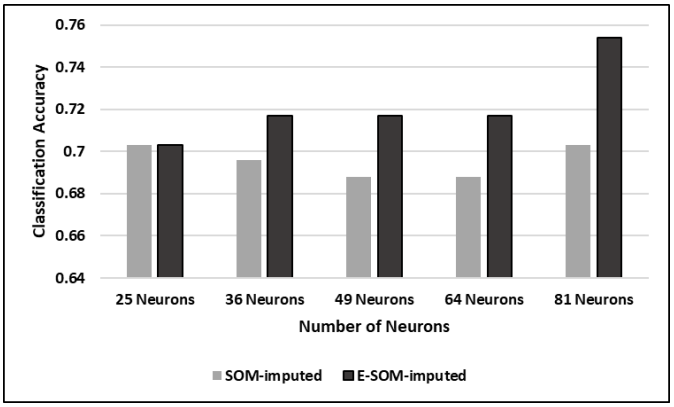

Figure 3: Testing Data Classification Accuracy Comparison Chart

Furthermore, on the E-SOM imputation method, for each variation of the number of neurons, several variations of the number of ensembles were applied. Variations in the number of ensembles used are 2, 5, and 10, resulting 15 complete training data. Figure 3 shows the accuracy values on testing data for each model built from E-SOM-imputed data, where in the E-SOM method, several variations of ensemble numbers were used. From Figure 4 it can be seen that for testing data classification, most optimal classification accuracy is obtained by the model built from E-SOM-imputed data using combination of 81 neurons and 15 ensembles.

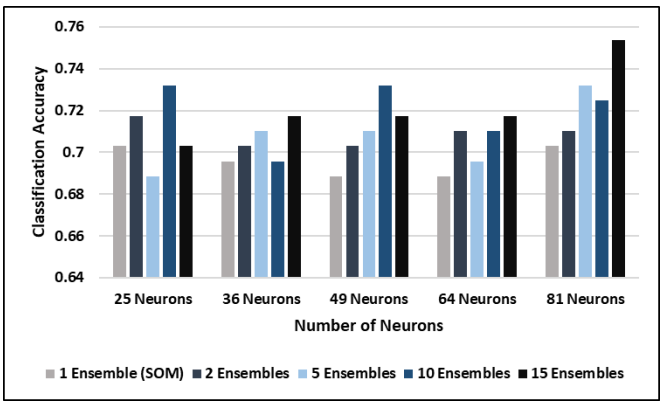

Figure 4: Testing Data Classification Accuracy with Variations of Ensemble Number

4. Conclusion

In this study, SOM and E-SOM methods have been implemented to impute missing values on South African Heart Disease data. Furthermore, a Random Forest model has been formed on the data from the imputation of the two methods. The results of the model evaluation showed that for accuracy in testing data, the Random Forest model formed from E-SOM imputed data yields better accuracy values than the Random Forest model formed from SOM-imputed data for variations of 36, 49, 64, and 81 neurons, while for variation of 25 neurons both models produce the same accuracy value. From the variation of the number of ensembles applied, the E-SOM imputation method with a combination of 81 neurons and 15 ensemble numbers produced a Random Forest model with the most optimal value of accuracy.

Acknowledgment

This research was supported by PITTA B 2019 Research Grant from Universitas Indonesia (ID Number: NKB-0677/UN2.R3.1/HKP.05.00/2019).

References

Gareth, J., Daniela, W., Trevor, H., & Robert, T. (2013). An introduction to statistical learning: with applications in R. Spinger.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1): 5–32. https://doi.org/10.1023/A:1010933404324

Bustamam, A., Rivai, M., & Siswantining, T. (2018). Implementation of spectral clustering on microarray data of carcinoma using self organizing map (SOM). *AIP Conference Proceedings*, *2023*(1), 020240. AIP Publishing LLC.

Cottrell, M., & Letrémy, P. (2007). Missing values: processing with the Kohonen algorithm. *ArXiv Preprint Math/0701152*.

Crambes, C., & Henchiri, Y. (2019). Regression imputation in the functional linear model with missing values in the response. *Journal of Statistical Planning and Inference*, *201*: 103–119.

Folguera, L., Zupan, J., Cicerone, D., & Magallanes, J. F. (2015a). Self-organizing maps for imputation of missing data in incomplete data matrices. *Chemometrics and Intelligent Laboratory Systems*, *143*: 146–151.

Friedman, J., Tibshirani, R., & Hastie, T. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer-Verlag New York New York.

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, *64*(5): 402.

Köhler, A., Ohrnberger, M., & Scherbaum, F. (2010). Unsupervised pattern recognition in continuous seismic wavefield records using self-organizing maps. *Geophysical Journal International*, *182*(3): 1619–1630.

Kubat, M. (1999). Neural networks: a comprehensive foundation by Simon Haykin, Macmillan, 1994, ISBN 0-02-352781-7. . *The Knowledge Engineering Review*, Vol. 13, pp. 409–412. https://doi.org/10.1017/s0269888998214044

Maimon, O. Z., & Rokach, L. (2014). *Data mining with decision trees: theory and applications* (Vol. 81). World scientific.

Nakagawa, S. (2015). Missing data: mechanisms, methods and messages. *Ecological Statistics: Contemporary Theory and Application*, 81–105.

Nishanth, K. J., & Ravi, V. (2016). Probabilistic neural network based categorical data imputation. *Neurocomputing*, *218*: 17–25.

Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv Preprint ArXiv:2010.16061*.

Rustum, R., & Adeloye, A. J. (2007). Replacing outliers and missing values from activated sludge data using Kohonen self-organizing map. *Journal of Environmental Engineering*, *133*(9): 909–916.

Saitoh, F. (2016). An ensemble model of self-organizing maps for imputation of missing values. *2016 IEEE 9th International Workshop on Computational Intelligence and Applications (IWCIA)*, 9–14. IEEE.

Verikas, A., Vaiciukynas, E., Gelzinis, A., Parker, J., & Olsson, M. (2016). Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness. *Sensors*, *16*(4): 592. https://doi.org/10.3390/s16040592

Yadav, M. L., & Roychoudhury, B. (2018). Handling missing values: A study of popular imputation packages in R. *Knowledge-Based Systems*, *160*: 104–118.

Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.