How to select a suitable machine learning algorithm: a feature-based, scope-oriented selection framework

Sala R.*, Zambetti M.*, Pirola F.*, Pinto R.*

**Department of Management, Information and Production Engineering, University of Bergamo, Viale Marconi, 5, Dalmine (BG), 24044, Italy (roberto.sala@unibg.it, michela.zambetti@unibg.it, fabiana.pirola@unibg.it, roberto.pinto@unibg.it)*

Abstract: The increasing availability of data gatherable from various sources and in several contexts, is forcing practitioners to find affordable ways to manage and exploit datasets. Within this context, machine learning (ML) - which can be described as a set of algorithms to analyse and process data to extract relevant features for clusterization, classification or prediction - emerged as one of the most investigated area providing powerful tools. Indeed, in literature it is possible to find a considerable number of articles dealing with ML algorithms and describing their real-world applications. This considerable number of works, depicting a wide variety of algorithms and widespread applications, creates an extensive knowledge on the topic. At the same time, it may also generate disorientation in the selection of the right approach. Thus, the need of synthesis and guidelines to drive the selection of the most suitable algorithm for a specific scope arises. To provide a response to such a necessity, the authors propose a ML algorithm selection tool. As a starting point, authors analysed several ML algorithms investigating their scope, their characteristics, and their typical fields of application, including also real examples. According to this exploration, authors identified two decision layers: the first one concerns the nature of the learning activity (supervised, unsupervised, etc.) while the second one is related to the characteristics of the ML algorithms (type of response, data size and type they can manage, etc.). Starting from a pool of algorithms, the first layer enables the users to narrow this pool depending on their scope. Then, the second layer guides the final selection, fitting the users' constraints, the previously mentioned algorithms features, and the data characteristics.

Keywords: machine learning; classification; selection framework; data analysis; decision making

1.Introduction

Nowadays, many companies are moving along the path toward digitisation in the light of the benefits envisaged by the 4th Industrial revolution (Coreynen et al., 2015; European Commission, 2016). One of the effects of the digitisation path is an unprecedented growth in data generation and availability. Still, it is clear that data availability per se does not lead to the creation of new value for companies, which can be generated only transforming data into useful information. For this reason, researchers and practitioners are struggling to find the correct way to manage data and extract value from them. The increased computational capabilities of modern PCs renewed the interest of research and industrial community around machine learning (ML), which refers to a set algorithms for analysing and process data for clusterization, classification or prediction purposes (Smola and Vishwanathan, 2008). Consequently, many researchers and practitioners are now approaching the ML field aiming at using these algorithms to extract value from their data. Literature reports applications of ML in the most various fields and contexts due to its flexibility and adaptability. However, a unique ML algorithm able to deal with the analysis of every dataset does not exist, making the selection of the right ML algorithm challenging (Andreopoulos et al., 2009; Kotsiantis et al., 2007; Saxena et al., 2017). Thus, this paper proposes a selection framework suggesting a set of algorithms able to deal properly with an analysed dataset. Rather than an instrument that selects the best algorithm available for the data analysis, this framework should be intended as a tool that supports the user discarding wrong algorithms for the data analysis.

Some of the frameworks currently available tend to use as main selection drivers the accuracy and/or prediction speed of the ML algorithms. An example of this is the "Machine Learning Algorithm Clean Sheet" proposed by Microsoft (Microsoft, 2017). The problem in this class of selection frameworks is that they may result too general most of the time. This means that the users are guided toward their choice without considering parameters like the algorithms' scalability or robustness to outliers, focusing only on the outcome (e.g. algorithm accuracy) and/or the amount of time required to obtain it. In this way, unsuitable ML algorithms may be selected leading in turn to poor results. In parallel, other selection approaches, built on different principles, are described in literature, like the Akaike Information Criterion (AIC) (Akaike, 1974), the Bayesian Information Criterion (BIC) (Schwarz, 1978) and the Hannan-Quinn Criterion (HQC) (Hannan and Quinn, 1979). These criteria consist in the computation of a specific value able to suggest the best statistical model to be used in an analysis. The problem with these criteria is that they require that the user knows

the characteristics of each model and that s/he computes the value for each single model under consideration. In this case, the model selection could be biased by the initial pool of algorithms selected by the user that could not include the most suitable ones.

The definition of the validation strategy for the proposed framework requires a detailed discussion due to the fact that the problem under study is non-trivial. The validation strategy should be developed considering the characteristics of the traditional methods and of the new framework, making possible to identify the pro and cons of each one and, in turn, to understand when one perform better than the others. For this reason, the framework validation is postponed to another paper.

The paper is structured as follows: Section 2 introduces the concepts of ML, discussing the main characteristics of the Supervised and Unsupervised learning approaches, and presenting a set of Supervised and Unsupervised algorithms. Section 3 deals with the definition of the framework, describing the drivers composing it and how they should be applied to datasets. Section 4 concludes the paper discussing the main benefits and limits of the framework.

2.Machine Learning

Mishra and Gupta (2017) defines ML as algorithms that, through the automatic association of events and their consequences, allows making accurate predictions based on a database of past observations.

Literature distinguishes Active Learning approaches from Passive Learning approaches. The main idea behind Active Learning is that it is a learning process where the ML algorithm is allowed to select the data from which it learns and, by that, is able to perform better with less training (Liu, 2010). On the contrary, the Passive Learning refers to an ensemble of approaches, namely Supervised, Unsupervised and Semi-Supervised, which use random samples from the dataset to build models that can perform prediction, classification, clustering or other tasks depending on the dataset composition (Mishra and Gupta, 2017). The differences among the three Passive Learning approaches is based on the presence of labels in the dataset.

Labels are constituted by one or more tags that contains desirable information on the data and favour its recognition. Thus, data can be classified as labelled or unlabelled. In particular, if the algorithm is trained on the base of labelled data the approach is identified as "Supervised Learning"; otherwise, in case of unlabelled data, the approach is called "Unsupervised Learning". In addition, if the training dataset contains both labelled and unlabelled data, the approach is called "Semi-Supervised Learning". The presented framework deals only with Supervised and Unsupervised learning approaches.

2.1 Supervised Learning

In the supervised learning, the idea is to exploit the information emerging from the data distribution and from the external knowledge – the labels – to create a model able to make predictions. Supervised learning approaches can be used for regression or classification purposes. In the first case, the idea is to use past data to build a regression model able to predict the future behaviour of the dataset. In the second case, the aim of the model is to classify data in specific classes, based on that, assign new data to the correct class. In essence, the supervised learning process aims to construct a mapping function conditioned to the provided training data set (Christiano Silva and Zhao, 2016). The following list reports the algorithms considered for classification purposes:

- Logit: this algorithm is suited for binary classifications. Logistic regression algorithm calculates the class membership probability for one of the two categories in a dataset. It is best suited for data clearly separated by a single, linear boundary (Dreiseitl and Ohno-Machado, 2002; Smola and Vishwanathan, 2008);
- Multinomial Logit: in the multinomial situation, there are different categorical response variables that can assume more than two outcomes. It basically gives an estimation about the class probabilities for a multi-category response which are then used to classify the new cases into one of several outcome groups (Dreiseitl and Ohno-Machado, 2002; Smola and Vishwanathan, 2008);
- Classification Trees: it is a non-parametric algorithm. For this reason, its performance is not affected by the presence of outliers. Even though it can handle a wide variety of input data, this algorithm is not suitable for high dimensional datasets. It has a flow chart structure, where each node represents a test and each leaf represent the response. It is easy to visualize, and results are easy to interpret (Singh et al., 2016);
- Support Vector Machines (SVM) Classification: the algorithm aims at classifying data by finding linear decision boundary (called hyperplane) which separates the data classes. The algorithm aims at finding the hyperplane that has the largest margin between two classes. For non-linear situations, the algorithm considers a loss function that penalizes the points on the wrong side of the hyperplane. Sometime this algorithm uses a kernel to transform nonlinearly separable data into higher dimensions where a linear decision boundary can be found. The SVM is suitable for binary data, but also discrete data can be used as input. High dimensional data can be managed easily. The algorithm performance decreases in presence of noise (Kotsiantis et al., 2007);
- k-Nearest Neighbour: this algorithm categorizes an object depending on the classes of the nearest neighbours in the dataset. Consequently, the algorithm assumes that objects that are close to each other are similar. The algorithm can be trained using different distance metrics (e.g.

Euclidean, Chebyshev, etc.). The algorithm can work with binary and discrete variables, but its performance is strongly affected by the data size and the presence of outliers and noise (Kotsiantis et al., 2007);

- (Multilayer) Neural Network: this algorithm consists of a set of simple, interconnected computation units called neurons, organized into layers with different roles called input, output and hidden layer, respectively. The number of hidden layers depends upon the model complexity. The neurons are connected via weighted links, and the way the neurons are connected defines different types of Neural Network. A Neural Network is trained iteratively to find the right weights for links. It best fits the modelling of highly nonlinear systems, when data are available incrementally and there is a constant need to update the model. Neural Networks algorithm can deal with noise and outliers in the dataset (Singh et al., 2016);

The following list reports the algorithms considered for regression purposes:

- Regression Trees: unlike the Classification Trees, Regression Trees can handle categorical and continuous variables. This algorithm is suitable when data has many features interacting in complicated and nonlinear ways. It sub-divides the space into smaller regions and further partitions the sub-divisions and assigns to its nodes (leaves) where interactions are more manageable. As for classification trees, nodes are subdivided into leaf nodes which contain the responses (Yildiz et al., 2017);
- SVM Regression: this kind of regression is very similar to a classification algorithm, but it is thought to predict a continuous response. It does not find a hyperplane to separate data, but it searches for a model that deviates from the measured data by a value that is not greater than a small amount, having the values of parameter as small as possible, in order to minimize the sensitivity to error. It is usually used for high-dimensional data, where there is a large number of predictor variables (Yildiz et al., 2017);
- k-Nearest Neighbour: it can be used in case of continuous data labels. The value of the parameter k influences the prediction variance: when it is small there is a high variance in prediction, while when it is high there is a large bias. The scale of features for KNN regression influences the quality of predictions (Hidalgo et al., 2017);
- (Multilayer) Neural Network: this algorithm is similar to the one used for classification purposes. The main difference with the classification versions of Neural Network is that, while in the first case the output is constituted by a discrete value (the class), here the output is constituted by a continuous value. Neural Networks algorithm can deal with noise and outliers in the dataset (Singh et al., 2016).

2.2 Unsupervised Learning

In the unsupervised learning case, the main task consists in finding intrinsic data structures. The learning process, in this case, is solely guided by the data relationships as no labels are available for the data analysis (Mitchell, 1997). The following list reports the algorithms considered for clustering purposes:

- Fuzzy C-Means (FCM): it uses fuzzy algorithms to handle data and analyse them. It is useful when data point can belong to more than one cluster. The number of clusters should be known. It is not suitable for datasets with noise or outliers but can handle large datasets (Havens et al., 2012);
- Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH): it is a hierarchical algorithm that takes as input a set of N data points and a desired number of clusters k. BIRCH relies on the use of clustering feature (CF) vectors to store and summarize the information about each cluster. It organizes these vectors in a CF tree, which is a height-balanced tree data structure. It is suitable for large datasets and is robust to outliers and noise. Disadvantages include a difficulty in finding arbitrary shaped clusters (Pitolli et al., 2017);
- Clustering Using Representatives (CURE): it is an improvement of the BIRCH algorithm since it is possible to find clusters of arbitrary shapes. CURE is also more robust with respect to outliers and scalable to large datasets. These benefits are achieved by using several representative objects for a cluster. At each iteration, the two clusters with the closest pair of representative objects are merged. A drawback is the user-specified parameter values, the number of clusters and the shrinking factor (Guha et al., 1998);
- RObust Clustering using linKs (ROCK): it assumes a similarity measure between objects and defines a 'link' between two objects whose similarity exceeds a threshold. Initially, each object is assigned to a separate cluster. Then, clusters are merged repeatedly according to their closeness. This algorithm is not able to handle properly large datasets and is not robust to outliers (Guha et al., 2000);
- k-Means: it divides data into k mutually exclusive clusters. The distance from the cluster centre defines the probability to belong to it. It fits large datasets but its performance decreases when outliers are present in the dataset (Pham et al., 2005);

- k-Medoids or Partitioning Around Medoids (PAM): it is similar to k-Means. The term k represents the number of medoids to be identified and the number of clusters that are required. A medoid is an object whose average dissimilarity from the other objects in the cluster is minimal and for this reason it is used as representative object (the centre) of the cluster. It processes and assigns n data points to k clusters with k medoids. In contrast to k-Means algorithm, k-Medoids uses a data in the cluster as cluster centre, while in k-Means the centroid is calculated and may not coincide with a data-point in the cluster. This makes the k-Medoids algorithm more robust in handling noise and outliers because it minimizes a sum of pairwise differences instead of a sum of Euclidean distances (Shafiq and Torunski, 2016).

Based on these definitions, the framework is proposed in the following section.

3.Framework

In the selection of a suitable ML algorithm for data analysis many aspects should be taken into account, and most of the times selecting an algorithm only on the base of the promised accuracy or computational speed leads to unsatisfactory results (Andreopoulos et al., 2009; Kotsiantis et al., 2007; Saxena et al., 2017). Moreover, the presence in literature of a wide range of different algorithms can create disorientation in the users not acquainted with their known strengths and weaknesses. This paper proposes a selection framework that works on two different layers, each one linked to a different aspect of the analysis. This would guide the user in the selection of the ML algorithms suitable for the analysis of a specific dataset. The first layer of the ML algorithm selection is based on the presence of labels in the dataset and on the scope of analysis (i.e. learning activity). In this way, the user is guided towards Supervised or Unsupervised Learning algorithms. Then, in case of a Supervised approach, the user is requested to indicate whether s/he is interested in a classification or a regression analysis. Otherwise, in case of Unsupervised Learning only clustering is proposed. In the second layer, four more drivers guide the user in the identification of proper ML algorithms. The drivers have been identified after a literature review of ML application cases. The results are reported in Table 1, which contains the list of drivers, their description and the list of papers used for their identification. It is worth to notice that some papers are associated to multiple drivers, in some cases to every driver. This underpins the importance of these drivers, strengthening the fact that these should be taken into consideration during the ML algorithm selection phase.

Table 1: Drivers Identified from Literature

Driver	Description	Literature
Data Type	The ML algorithms are built under specific assumptions, each one is usually informed to work with specific types of data. Here, four types of data are considered: binary, discrete, categorical, and continuous. Discrete data can assume only certain values and are always numeric (e.g. number of students, etc…); categorical data can assume only certain value from a finite and fixed set (e.g. colours, months in a year, etc…); continuous data can assume any value in an interval (e.g. height, time to complete a task, etc…).	(Andreopoulos et al., 2009; Dreiseitl and Ohno-Machado, 2002; Guha et al., 2000; Kotsiantis, 2007; Pham et al., 2005; Pitolli et al., 2017; Saxena et al., 2017; Singh et al., 2016)
Scalability	The scalability of an algorithm measures the growth of its time complexity in relation to the growth of the problem size. It measures the capacity of an algorithm to handle big inputs (Teng, 2018). In the proposed framework, a ML algorithm is scalable if it is able to deal with both small datasets (few features) and large datasets (many features) without decreasing its performance. For example, the runtime and memory requirements should not increase exponentially when increasing the number of features in the datasets.	(Andreopoulos et al., 2009; Dreiseitl and Ohno-Machado, 2002; Guha et al., 1998, 2000; Havens et al., 2012; Hidalgo et al., 2017; Kotsiantis, 2007; Pham et al., 2005; Pitolli et al., 2017; Saxena et al., 2017; Shafiq and Torunski, 2016; Singh et al., 2016; Teng, 2018)
Robustness to Outliers	It is defined as the ability of a ML algorithm to deal with the presence of data not belonging to the analysed sample. If an algorithm is robust to outliers and noise, its performance is not affected by their presence.	(Andreopoulos et al., 2009; Guha et al., 1998, 2000; Havens et al., 2012; Kotsiantis, 2007; Pitolli et al., 2017; Saxena et al., 2017; Shafiq and Torunski, 2016; Singh et al., 2016)
Response Type	It is defined as the outcome of the analysis. As for the Data Type driver, there are different possible types of response for the analysis. As for the Data Type driver, the possible Response Types are binary, discrete, categorical and continuous.	(Dreiseitl and Ohno-Machado, 2002; Guha et al., 1998, 2000; Kotsiantis, 2007; Pham et al., 2005; Pitolli et al., 2017; Saxena et al., 2017)

Figure 1 depicts the two layers of the framework and the related selection drivers, while Table 2 shows the list of algorithms classified based on those drivers. As explained earlier, following this framework the users should be able to identify one or more ML algorithms suitable for their scope (drivers Learning and Learning Activity) and the dataset characteristics (drivers Data Type, Scalability, Robustness to Outliers/Noise and Response Type).

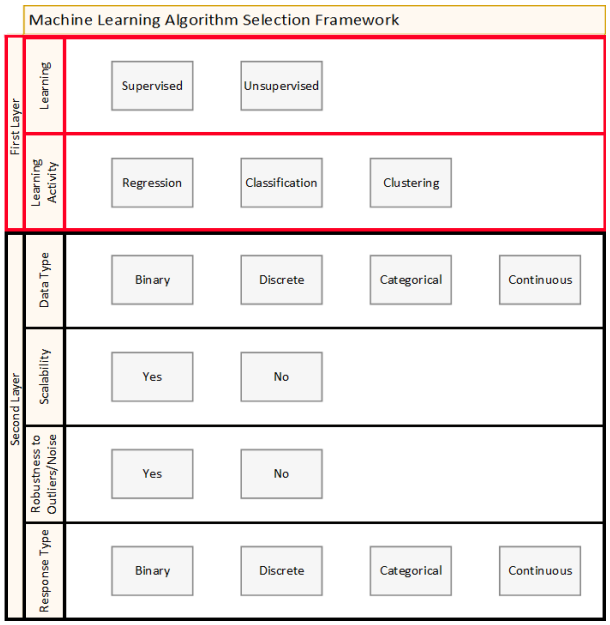

Figure 1: Machine Learning Algorithm Selection Framework

In order to be effective, the proposed selection framework has to be applied in a precise sequence. In particular, at the beginning, the first layer requires to select the learning approach and, in turn, the learning activity. Then, the second selection layer requires the identification of the data characteristics to understand which of the available algorithms should be used for the analysis. As above mentioned, the user should consider the nature of the data in order to avoid algorithms unable to deal with the dataset. Thus, knowing the dataset dimension, the user is required to indicate whether the it has a limited number of features, and so can be handled by most of the algorithms, or requires specific algorithms, whose performance are not affected by the number of features. Moreover, the user should consider the possibility to have outliers or noise in the dataset and clarify if their presence is a problem or not for the analysis. If it is a problem, the framework removes the algorithms not able to manage the presence of outliers and/or noise. Finally, the framework requires the user to specify the type of response required as output of the analysis.

4.Conclusions

This paper presented a selection framework aimed at guiding users in the selection of one or more suitable ML algorithms to use in their analyses. The proposed selection framework does not claim to be complete with respect to the available literature on the topic and/or to suggest the best algorithm to the user due to the vast complexity characterising the ML field of research and application. Moreover, this paper deals only with the framework development but not with its validation due to the complexity of the problem and due to the space limitations.

This selection framework aims at supporting the researchers who are approaching the ML field guiding them through the selection of a set of algorithms suitable for their scopes, helping them to avoid incurring in the application of algorithms which are not suitable for the data they are dealing with. The selection framework covers some of the most commonly used ML Supervised and Unsupervised algorithms. The drivers presented in the selection framework constitute a solid base for the selection of proper ML algorithms since they are easily recognizable and do not necessitate deep analysis to be identified.

The research presented in this paper is not free from limitations and possible future improvements. First, the current pool of ML algorithms lacks Semi-Supervised and Active Learning approaches. Furthermore, more classification algorithms, as well as more regression algorithms, could be considered in the framework. Examples of these algorithms are the generalized linear models, the Bayesian networks, the linear and quadratic discriminant analysis, the gaussian processes etc. Also, the pool of unsupervised clustering algorithms could be extended including new algorithms such as the affinity propagation, the spectral clustering, the gaussian mixtures, the agglomerative clustering etc. Moreover, the level of detail used for each type of algorithm could be improved (e.g. the variants of Classification Trees, of Neural Networks, the different kernels available for the SVM etc.). Second, the drivers list could be extended considering other characteristics of the data such as the training time, the required parameters etc. Third, due to the number of algorithms currently available in the framework and the possible combinations of the dataset characteristics, in some cases the framework may be unable to suggest a suitable ML algorithm. Fourth, the framework currently does not take into consideration the application of data manipulation techniques that could modify the dataset structure in the pre-processing phase and, in turn, extend the pool of algorithms to be taken into account for the analysis.

Future work will encompass multiple aspects not considered in this paper, starting from the validation strategy, which should be chosen carefully to be effective, and continuing with the selection framework extension and refinement.

Table 2: Machine Learning Algorithm Classification

ML Algorithm	First layer		Second layer			
	Learning Type	Learning Activity	Data Type	Scalability	Robustness to Noise/Outliers	Response Type
Logit	Supervised	Classification	Binary	Yes	Yes	Binary
Multinomial Logit	Supervised	Classification	Discrete – Categorical	Yes	Yes	Categorical
Decision Trees	Supervised	Classification	Binary - Discrete - Categorical	No	Yes	Binary – Discrete - Categorical
SVM Classification	Supervised	Classification	Binary - Discrete	Yes	No	Binary - Discrete
K-Nearest Neighbour	Supervised	Classification	Binary - Discrete	No	No	Binary - Categorical
Neural Networks	Supervised	Classification	Binary - Discrete - Categorical	Yes	Yes	Binary - Categorical
Regression Trees	Supervised	Regression	Discrete - Categorical - Continuous	No	Yes	Categorical-Continuous
SVM Regression	Supervised	Regression	Discrete - Continuous	Yes	No	Discrete - Continuous
K-Nearest Neighbour	Supervised	Regression	Discrete - Continuous	No	No	Categorical
Neural Networks	Supervised	Regression	Binary - Discrete - Categorical - Continuous	Yes	Yes	Categorical - Continuous
k-Mean	Unsupervised	Clustering	Continuous	Yes	No	Continuous - Categorical
k-Medoids (PAM)	Unsupervised	Clustering	Continuous	Yes	Yes	Categorical
FCM (Fuzzy C-Means)	Unsupervised	Clustering	Continuous	Yes	No	Categorical
BIRCH	Unsupervised	Clustering	Continuous	Yes	Yes	Discrete
CURE	Unsupervised	Clustering	Continuous	Yes	Yes	Discrete
ROCK	Unsupervised	Clustering	Continuous	No	Yes	Discrete

Acknowledgements

This work was funded by Lombardy Region - FESR 2014-2020 innovazione e competitività "Bando Linea R&S per aggregazioni" in the project "Proactive Maintenance and rEal Time monitoring for Efficiency & Ø defect production (PROMETEØ)", project ID: 148633, CUP: E47H16001570009.

References

Akaike, H. (1974), "A New Look at the Statistical Model Identification", *IEEE Transactions on Automatic Control*, Vol. 19 No. 6, pp. 716–723.

Andreopoulos, B., An, A., Wang, X. and Schroeder, M. (2009), "A roadmap of clustering algorithms: Finding a match for a biomedical application", *Briefings in Bioinformatics*, Vol. 10 No. 3, pp. 297–314.

Christiano Silva, T. and Zhao, L. (2016), *Machine Learning in Complex Networks*, *Book*, Springer.

Coreynen, W., Matthyssens, P. and Van Bockhaven, W. (2015), "Boosting servitization through digitization: Pathways and dynamic resource configurations for manufacturers", *Industrial Marketing Management*, Elsevier Inc., Vol. 60, pp. 42–53.

Dreiseitl, S. and Ohno-Machado, L. (2002), "Logistic regression and artificial neural network classification models: A methodology review", *Journal of Biomedical*

Informatics, Vol. 35 No. 5–6, pp. 352–359.

European Commission. (2016), "Digitising European Industry Reaping the full benefits of a Digital Single Market".

Guha, S., Rastogi, R. and Shim, K. (1998), "CURE: an efficient clustering algorithm for large databases", *ACM SIGMOD Record*, Vol. 27 No. 2, pp. 73–84.

Guha, S., Rastogi, R. and Shim, K. (2000), "Rock: a robust clustering algorithm for categorical attributes", *Information Systems*, Vol. 25 No. 5, pp. 345–366.

Hannan, E.J. and Quinn, B.G. (1979), "The Determination of the Order of an Autoregression", *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 41 No. 2, pp. 190–195.

Havens, T.C., Bezdek, J.C., Leckie, C., Hall, L.O. and Palaniswami, M. (2012), "Fuzzy c-Means algorithms for very large data", *IEEE Transactions on Fuzzy Systems*, Vol. 20 No. 6, pp. 1130–1146.

Hidalgo, J.I., Colmenar, J.M., Kronberger, G., Winkler, S.M., Garnica, O. and Lanchares, J. (2017), "Data Based Prediction of Blood Glucose Concentrations Using Evolutionary Methods", *Journal of Medical Systems*, Vol. 41 No. 9.

Kotsiantis, S.B. (2007), "Supervised machine learning: a review of classification techniques.", *Informatica (03505596)*, Vol. 31 No. 3, pp. 249–268.

Kotsiantis, S.B., Zaharakis, I. and Pintelas, P. (2007), "Supervised machine learning: A review of classification techniques", Vol. 31, pp. 249–268.

Liu, Y. (2010), "Active learning literature survey", *Computer Sciences Technical Report 1648*.

Microsoft. (2017), "The Machine Learning Algorithm Cheat Sheet", available at: https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-choice#the-machine-learning-algorithm-cheat-sheet.

Mishra, C. and Gupta, D.L. (2017), "Deep Machine Learning and Neural Networks: An Overview", *IAES International Journal of Artificial Intelligence (IJ-AI)*, Vol. 6 No. 2, pp. 66–73.

Mitchell, T.M. (1997), *Machine Learning*, McGraw-Hill Science/Engineering/Math.

Pham, D.T., Dimov, S.S. and Nguyen, C.D. (2005), "Selection of K in K-means clustering", *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, Vol. 219 No. 1, pp. 103–119.

Pitolli, G., Aniello, L., Laurenza, G., Querzoni, L. and Baldoni, R. (2017), "Malware family identification with BIRCH clustering", *Proceedings - International Carnahan Conference on Security Technology*, Vol. 2017–Octob.

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., et al. (2017), "A review of clustering techniques and developments", *Neurocomputing*, Elsevier B.V., Vol. 267, pp. 664–681.

Schwarz, G. (1978), "Estimating the Dimension of a Model", *The Annals of Statistics*, Vol. 6 No. 2, pp. 461–464.

Shafiq, M.O. and Torunski, E. (2016), "A Parallel K-Medoids Algorithm for Clustering based on MapReduce", *2016 15th IEEE International Conference on Machine Learning and Applications A*, pp. 502–207.

Singh, A., Thakur, N. and Sharma, A. (2016), "A Review of Supervised Machine Learning Algorithms", *2016 International Conference on Computing for Sustainable Global Development (INDIACom)*, Vol. 16, pp. 1310–1315.

Smola, A. and Vishwanathan, S.V.N. (2008), *Introduction to Machine Learning*.

Teng, S.-H. (2018), "Scalable Algorithms in the Age of Big Data and Network Sciences", *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining - WSDM '18*, pp. 6–7.

Yildiz, B., Bilbao, J.I. and Sproul, A.B. (2017), "A review and analysis of regression and machine learning models on commercial building electricity load forecasting", *Renewable and Sustainable Energy Reviews*, Elsevier Ltd, Vol. 73 No. February, pp. 1104–1122.