International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

Training Subset Selection to Improve Prediction Accuracy in Investment Ranking

Mehmet Köseoğlu[1]

Accepted : 11/03/2019 Published: 31/03/2019

Abstract: Most studies in the supervised learning literature assume that the training set and the test set are generated from the same distribution. If this assumption does not hold, training a model on the whole dataset may significantly reduce prediction accuracy. Here we propose a training instance selection method which constructs a subset of the training set to maximize prediction accuracy. We have applied the proposed algorithm to an investment ranking problem where the training dataset consists of multiple time periods. Our algorithm finds the best group of periods to include in the training set to maximize prediction accuracy for a given target period. By only including similar periods to the training set, prediction performance is significantly improved against a training scheme which uses all of the previous periods to train the model. The proposed algorithm ranked first in the IEEE Investment Ranking Challenge 2018 which was organized as a part of the IEEE Data Science Workshop 2018.

Keywords: *Supervised learning, instance selection, concept drift, investment ranking*

1. Introduction

Most studies in the supervised learning literature assume that the training test sets are generated from the same distribution. The datasets commonly used in machine learning benchmarks are crafted such that the training and test sets are randomly selected from the same data source resulting in a homogeneous training/test split. While this assumption holds for benchmark datasets, there may be groups of training samples in real-life problems which do not correlate well with samples in the test set. In such a case, using unrelated samples in building the predictive model significantly reduces the prediction accuracy.

The problem of having a different distribution between the training and the test set can be named as dataset shift or concept drift [1] [2]. Such a shift can be encountered in many real-world situations: For example, a classifier trained on the data obtained from a laboratory can be used in another laboratory with different devices [3]. Similarly, an application which has adapted to the past behavior of a user may fail when the user's behavior changes [4]. Also, as in the case of application we consider here, the financial markets undergo changes where the models built using previous information may not accurately predict future returns. These type of shifts results in lower accuracy in regression or classification tasks. In that case, using the most relevant samples from the training set with respect to the shifted test set is crucial. Selecting a subset of samples from the training dataset is called as instance selection [5].

Here we propose a training instance selection method which eliminates the possible dissimilarity among the training and the test set. Given a heterogeneous training set which has some subsets more relevant to the test set, the proposed algorithm searches for the optimum subsets of the training data to maximize prediction accuracy. The proposed method uses the prediction accuracy over the validation set to guide the search through the whole training set to find the optimum subset of the training samples.

We apply the proposed technique to a stock market prediction problem where the aim is to predict the return of a stock using its features. The training set of the problem consists of features and returns of anonymous stocks for different time periods. The proposed algorithm finds the best group of periods to include in the training set to build a predictive model for a given target period.

The main motivation behind this application is that the market condition is the determining factor behind the relationship between a stock's features and its return. For example, the relationship between the features and the return of a stock may be significantly different for a year of global crisis in comparison to a year of high market returns. By including the most relevant periods to the target period into the training set, it is possible to improve prediction accuracy significantly. Assuming we have access to the performance of a part of the stocks for a given period, we can search through the previous periods to find the best periods to include in the training set.

Our algorithm obtained the best result in the IEEE Investment Ranking Challenge 2018 which was organized as a part of the IEEE Data Science Workshop[2]. When the best training subset found by the proposed algorithm is used in the prediction task, the average Spearman's coefficient for the target periods is 0.27 whereas it was approximately zero when the whole training data is used. In other words, without the proposed instance selection method, the prediction accuracy of the supervised learning algorithm is not better than random picking of stocks.

We present the related work in the next section. The proposed algorithm and its application to investment ranking are explained in Secs. 3 and 4, respectively. Secs. 5 and 6 present the discussion and conclusion.

2. Related Work

In this section, we present the related work from both the supervised learning literature and the financial data analysis literature.

[1] *Dept. Of Computer Engineering, Hacettepe University, Ankara - 06800, TURKEY. Email: mkoseoglu@cs.hacettepe.edu.tr*

Relevant previous studies from the supervised learning literature focus on either prototype selection or training set selection [5]. Prototype selection methods aims to improve the interpretability of a dataset or the scalability of data processing without a particular focus on training [6], [7]. The training set selection methods focus on selecting training samples to use in the training procedure [8], [9], [10]. The aim of the most previous studies is to reduce the size of the training dataset by removing unnecessary training samples to improve scalability. Instead, our main goal is to improve prediction performance. Most of the previous instance selection techniques are focused on classification tasks rather than regression [11] whereas we consider a regression problem. Moreover, the individual samples are considered in the previous instance selection studies; but, we consider a setting where distinct groups of samples can be found in the training set.

The proposed technique is also related to wrapper methods in feature selection. In these methods, the features which improve prediction accuracy is selected [12]. Here we select a subset of training examples which would lead to better prediction instead of features.

In the finance literature, there are studies which consider the discrepancy between the training and test sets due to the time changing behavior of markets. Su and Li investigated the financial distress prediction of companies using training instance selection [13]. Another study also deals with training set shaping for credit scoring [14]. Kim proposes a genetic algorithm-based instance selection method for stock market prediction [15].

The proposed work is also relevant to the literature on concept drift [16] and domain adaptation [17]. Concept drift is non-stationary behavior of data sources which can be encountered in many domains [18]. Domain adaptation deals with adapting the models learned from one source to a different but similar target source.

3. Method

In this section, we present the training subset selection algorithm which builds a training set to maximize prediction accuracy.

3.1. Problem definition

Let S_{train} be the training set of a supervised learning problem and $(x, y) \in S_{train}$ are the examples in the training set. The supervised learning objective is to find the optimum parameters $\hat{w}$ of the predictive function f such that the training loss function is minimized:

$$\hat{w} = \arg\min_{w} \mathcal{L}(w, \mathbb{S}_{\text{train}}) = \frac{1}{|\mathbb{S}_{\text{train}}|} \sum_{(x_i, y_i) \in \mathbb{S}_{\text{train}}} \mathcal{L}_i(x_i, y_i; w) \quad (1)$$

where $\mathcal{L}_i(x, y; w)$ is the loss function for sample i such as the squared error:

$$\mathcal{L}_i(x, y; w) = (y_i - f(x_i; w))^2. \quad (2)$$

The main assumption in the above formulation is that the training and test sets are generated from the same distribution. Hence, it is usually assumed that f which minimizes the error on the training set will minimize the error on the test set if overfitting is avoided.

If the training set and the test set is generated from different distributions, then, minimization of loss with respect to the training set may not minimize loss on the test set. Here we consider a case where there are subsets of the training set which correlates very well with the test set and there are some subsets which are redundant or even detrimental to the prediction task.

Let $\mathcal{P}(\mathbb{S})$ be all subsets of the training set $\mathbb{S}$. Our aim is to find the optimum subset $\hat{\mathbb{S}} \in \mathcal{P}(\mathbb{S})$ where the parameters of the prediction function optimized on this subset

$$\hat{w} = \arg\min_{w} \mathcal{L}(w, \hat{\mathbb{S}}) = \frac{1}{|\hat{\mathbb{S}}|} \sum_{(x_i, y_i) \in \hat{\mathbb{S}}} \mathcal{L}_i(x_i, y_i; w) \quad (3)$$

minimizes the loss function on the validation set.

Instead of considering individual samples from the training set, we assume that the training set, $\mathbb{S}$, consists of N disjoint subsets where $\mathbb{X}_i$ be the i^{th} subset, i.e. $\mathbb{S} = \cup_{i=0}^{N-1} \mathbb{X}_i$. In a typical application, these subsets may be the data collected from different groups of users or from different time periods as in the financial application that we consider here. It is also possible to obtain the subsets by using a clustering method where the training samples are grouped according to their similarity to each other.

Algorithm 1: Training Subset Selection Algorithm

$\mathbb{S} = \cup_{i=0}^{t-1} \mathbb{X}_i$

$a_{max} = 0$

$\mathbb{S}^{best} = \mathbb{S}$

repeat

  $\mathbb{S}^{last} = \mathbb{S}^{best}$

  for i<N do

    if $\mathbb{X}_i \in \mathbb{S}$ **then**

      $\mathbb{S} = \mathbb{S} \setminus \mathbb{X}_i$

      excluded = True

    else

      $\mathbb{S} = \mathbb{S} \cup \mathbb{X}_i$

      excluded = False

    end if

    Train a predictor using $\mathbb{S}$

    Obtain a prediction accuracy, a

    if $a > a_{max}$ **then**

      $a_{max} = a$

      $\mathbb{S}^{best} = \mathbb{S}$

    else

      if excluded == True **then**

        $\mathbb{S} = \mathbb{S} \cup \mathbb{X}_i$

      else

        $\mathbb{S} = \mathbb{S} \setminus \mathbb{X}_i$

      end if

    end if

    i = i + 1

  end for

until $\mathbb{S}^{last} == \mathbb{S}^{best}$

3.2. Training Subset Selection Algorithm

The proposed algorithm is presented in Algorithm 1. The algorithm starts with the complete training data $\mathbb{S} = \cup_{i=0}^{N-1} \mathbb{X}_i$ at the beginning. It trains the supervised learning model using this training set and obtains a validation error. It then removes the first subset, $\mathbb{X}_0$, from the training set and checks whether this exclusion improves validation error.

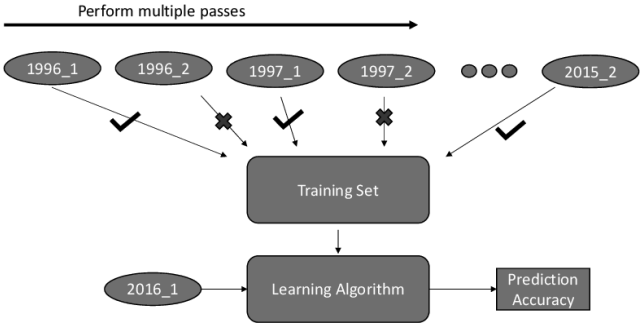


Fig. 1 - Illustration of the proposed algorithm for the investment ranking application that we consider.

If the exclusion improves accuracy, that subset remains excluded from the set. If not, it is kept in the training set. Then, the algorithm switches to the next subset, excludes it from the training set and similarly it is kept removed from the set if exclusion improves performance. The algorithm goes through all periods in the training set in a similar fashion.

After completing the first pass through all subsets, the algorithm starts over again. If a subset is excluded from the dataset in the first round, the algorithm includes it again to check if inclusion will improve the performance. If a period is included in the first round, it is excluded to check if exclusion improves the performance. The algorithm goes on searching for the optimum training dataset until the training set does not change over two successive iterations. The formal description of the procedure can be seen in Algorithm 1 and its illustration is given in Fig. 1.

The aim of this search is to find the optimum dataset which maximizes the prediction performance. Although this search procedure can be replaced by an exhaustive search which checks all possible combinations of subsets, its complexity would be 2^N. The proposed algorithm is a heuristic which approximates the optimum dataset in a computationally efficient way. Our results described in Sec. 4 indicates that the algorithm converges in few iterations for the application that we consider.

4. Application to Investment Ranking

In this section, we present the evaluation of the proposed algorithm on the dataset provided for the 2018 IEEE Investment Ranking Challenge[1]. The dataset consists of semi-annual returns for a group of stocks for 6-month periods from 1995 to 2017, i.e. there are 42 periods. There are approximately 900 stocks for each period and the stock identifiers are anonymized. For each month of a 6-month period, 71 anonymized features are given for each stock. The target periods are all periods starting from the first half of 2002 to the second half of 2016. The returns are normalized for each period whereas the features were normalized over all periods. 40% of the dataset is reserved for testing. Since the stock identifiers are not consistent among different periods, it is not possible to make a time-series analysis on the performance of a given stock.

4.1. Pre-processing

We have averaged each feature over the six months of a period resulting a single value for each feature per period. In the dataset, the features were normalized over all periods but returns were normalized for each individual period. We have further normalized the features for each period. We have also replaced the missing features for a given stock by using the mean value of that feature for all other stocks for that period. If a feature is missing for all stocks for a period, it is replaced with a zero.

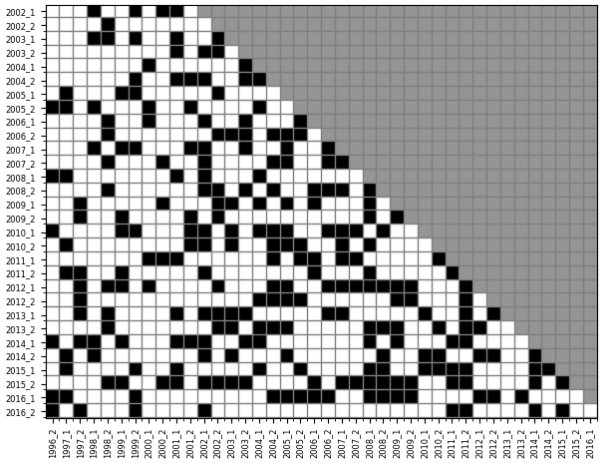


Fig. 2 - Periods included in the training set of a given year. Y-axis indicates the target period to be predicted and x-axis indicates the periods in the whole training set. Black points indicate that corresponding period is included in the training set.

4.2. Supervised Learning Algorithm

The training subset selection algorithm that we propose is not dependent on a specific type of supervised learning algorithm. In our experiments, we have obtained the best results using Bayesian linear regression [19]. In Bayesian regression, a Gaussian prior is assumed on the model parameters, i.e. $\mathbf{w} \sim \mathcal{N}(0, \lambda^{-1}\mathbf{I})$ and the output y is distributed as $\mathbf{y} \sim \mathcal{N}(\mathbf{Xw}, \alpha)$ where $\mathbf{X}$ is the training data. Due to the Gaussian prior, model parameters tend to be close to zero which provides regularization and prevents overfitting.

We have also obtained results using linear regression with l2 regularization, linear regression without regularization, linear regression with l1 regularization and artificial neural networks. The results of these experiments are presented in the next section.

We have used scikit-learn module of Python [20] to evaluate different regression models. The sckit-learn is an open-source library which provides tools for machine learning tasks such as regression, classification, and clustering. The consistent interface of the library allows easy experimentation using different techniques. We have used this capability of the library to decide on the supervised learning technique after trials with different approaches.

4.3. Numerical Results

We evaluate the performance of the proposed algorithm using Spearman's correlation. Spearman's correlation is the equivalent of the Pearson's correlation coefficient for rankings and it can be computed as follows for distinct integer rankings:

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \quad (4)$$

where n is the number of observations and d_i is the difference between the two ranks of i^{th} observation. A perfect correlation results in a Spearman's coefficient of 1, a perfect negative correlation results in -1 and a random picking of stocks results in a correlation coefficient around 0.

[1] The dataset is publicly available through the challenge web page: https://www.crowdai.org/challenges/ieee-investment-ranking-challenge/dataset_files

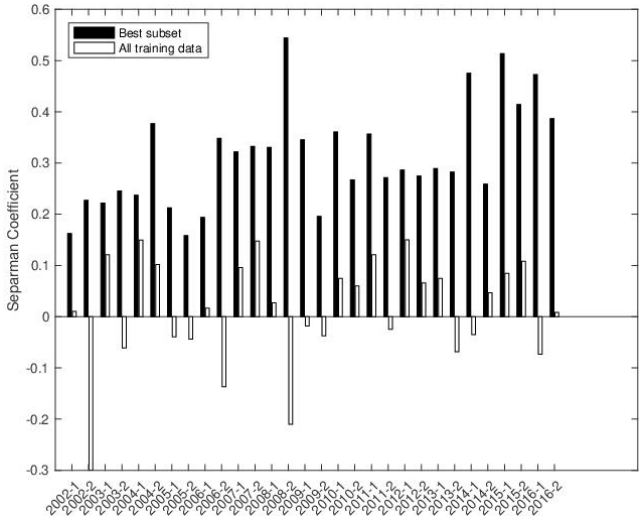


Fig. 3 - Spearman's coefficient for target periods. White bars indicate the prediction performance when all periods are included in the training set whereas black bars are for the training sets found by the proposed algorithm.

We have used Algorithm 1 as the training subset selection algorithm and experimented with different supervised learning algorithms. Table 1 summarizes the results obtained by different supervised learning algorithms when used together with the Algorithm 1. We have obtained the best results on the hidden test set in the competition using the Bayesian linear regression. In the Bayesian linear regression, the priors for α and λ are assumed to be Gamma distributed with hyper-parameters 10^{-6}. For the artificial neural network, we have assumed a single hidden layer with a size of 3 with tanh activations and trained using L-BFGS optimization. For linear regression with l1 and l2 regularization, we have used 10-fold cross-validation.

Table 1 also indicates the total computation time to generate predictions for all periods for different supervised techniques using a personal computer having a 3.5 GHz Intel Xeon CPU with 4 cores. Bayesian linear regression is the fastest of the methods we experimented.

Fig. 2 summarizes which periods are included in the training set by Algorithm 1 for a given prediction period. The black squares indicate that the period is included in the dataset. This figure also indicates the similarity between the characteristics of two periods. There is no visible significant correlation among the periods included in the training periods for different target periods, hence it is not possible to discern among different market regimes by investigating the training sets for individual periods.

Fig. 3 shows the prediction performance when all periods are included in the training set for a given period. The figure also includes prediction performance when the training set is the best subset found by the proposed algorithm. On the test set, the average Spearman correlation scores for our algorithm were 0.312. When the whole training data is used, the score is 0.014. These results suggest that without training subset selection, supervised learning algorithms can barely outperform random ranking of stocks and the proposed algorithm improves the prediction performance significantly.

When the whole training data is used, supervised learning performed worst for the periods of 2002_2 and 2008_2 resulting in a Spearman's coefficient of less than -0.2. Hence, for those two periods, a naive supervised learning approach performs much worse than a random picking of stocks. Given that those two periods are times of global financial crises, this may imply that a naive supervised learning approach based on the previous history should not be trusted at times of financial instability.

The computational performance of the proposed algorithm was quite good in our experiments. In the worst case, the number of iterations needed in Algorithm 1 was 5. In comparison to the 2^{40} number of iterations which would be required for an exhaustive search, the computational complexity is significantly reduced.

Table 1 – Prediction accuracy achieved by each supervised learning technique when Algorithm 1 is applied.

Supervised Learning Technique	**Spearman's Corr. Coef.**	**Computation time (sec)**
Bayesian Linear Regression	0.312	758
Linear Regression w/o regularization	0.311	771
Linear Regression with l1 regularization with 10-fold cv	0.243	2827
Linear Regression with l2 regularization with 10-fold cv	0.312	2144
Artificial Neural Network	0.168	2260

5. Conclusions

A fundamental assumption in supervised learning is that the training and tests are generated from the same distribution. This assumption may not hold for practical scenarios when the streaming data distribution shifts as time progresses. This is especially true for financial systems where the relationship between the features and returns of a stock strongly depends on the general market conditions. In such cases, forming a training subset resembling the test set is crucial. Our results suggest that the proposed training subset selection algorithm significantly improves prediction accuracy when there is significant heterogeneity among the training set and the test set.

We have proposed a training subset selection method for supervised learning problems and applied it to an investment ranking problem. Proposed technique improved the prediction accuracy significantly with respect to a naive learning approach where whole dataset is used in training. Our results suggest that training subset selection can be a promising technique for supervised learning especially for problems where the training and test datasets are generated from different distributions.

The proposed technique can be considered similar to wrapper methods of feature selection where the features are selected based on the prediction performance. Although there is a rich literature on feature selection, there are not many studies dealing with training instance selection except relatively few studies focusing on performance improvements. Our results suggest that there is a potential for improvement in training instance selection. It should be noted that, however, the complexity of training instance selection is much higher than feature selection as there are many more samples than features in a typical dataset. As in our application, having distinct groups of training samples reduces the complexity of the search.

With regards to the investment ranking problem that we consider, the generalization of the proposed method to future periods is not trivial. If there is no validation set for a target period, it is not possible to use validation accuracy to form the training subset. A future possible research direction can be to partition the training periods into a few clusters and match those clusters to some of the features of the target period. Then, for a given target period, it may be possible to find the training set based on its features.

Acknowledgements

This work is done when Mehmet Koseoglu was a visiting postdoctoral researcher at UCLA. His visit is supported by the Fulbright Commission with grant number FY-2017-TR-PD-02. He would like to thank Ayca Ozcelikkale from Uppsala University for helpful discussions during the competition and her comments on the manuscript. He also would like thank Mani Srivastava from UCLA for hosting him.

References

[1] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964-994, 2016.

[2] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer and N. D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.

[3] J. G. Moreno-Torres, X. Llorà, D. E. Goldberg and R. Bhargava, "Repairing fractures between data using genetic programming-based feature extraction: A case study in cancer diagnosis," *Information Sciences*, vol. 222, pp. 805-823, 2013.

[4] Y. Lo, W. Liao, C. Chang and Y. Lee, "Temporal Matrix Factorization for Tracking Concept Drift in Individual User Preferences," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 156-168, 2018.

[5] S. Garcia, J. Luengo and F. Herrera, "Instance Selection," in *Data Preprocessing in Data Mining*, Springer, 2015, pp. 195-243.

[6] J. Bien and R. Tibshirani, "Prototype Selection for Interpretable Classification," *The Annals of Applied Statistics*, vol. 5, pp. 2403-2424, 2011.

[7] F. Zhu, B. Fan, X. Zhu, Y. Wang, S. Xiang and C. Pan, "10,000+ Times Accelerated Robust Subset Selection (ARSS)," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, 2015.

[8] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, J. F. Martinez-Trinidad and J. Kittler, "A review of instance selection methods," *Artificial Intelligence Review*, vol. 34, pp. 133-143, 01 8 2010.

[9] E. Leyva, A. Gonzalez and R. Perez, "Three new instance selection methods based on local sets: A comparative study with several approaches from a bi-objective perspective," *Pattern Recognition*, vol. 48, pp. 1523-1537, 2015.

[10] B. Du, W. Z. L. Zhang, L. Zhang, W. Liu, J. Shen and D. & Tao, "Exploring representativeness and informativeness for active learning," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 14-26, 2017.

[11] Á. Arnaiz-Gonzalez, M. Blachnik, M. Kordos and C. Garcia-Osorio, "Fusion of instance selection methods in regression tasks," *Information Fusion*, vol. 30, pp. 69-79, 2016.

[12] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, pp. 245-271, 1997.

[13] J. Sun and H. Li, "Dynamic financial distress prediction using instance selection for the disposal of concept drift," *Expert Systems with Applications*, vol. 38, no. 3, pp. 2566-2576, 2011.

[14] M. Saberi, M. S. Mirtalaie, F. K. Hussain, A. Azadeh, O. K. Hussain and B. Ashjari, "A granular computing-based approach to credit scoring modeling," *Neurocomputing*, vol. 122, pp. 100-115, 2013.

[15] K.-j. Kim, "Artificial neural networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications*, vol. 30, no. 3, pp. 519-526, 2006.

[16] P. R. Almeida, L. S. Oliveira, A. S. Britto and R. Sabourin, "Adapting dynamic classifier selection for concept drift," *Expert Systems with Applications*, vol. 104, pp. 67-85, 2018.

[17] N. Courty, R. Flamary, D. Tuia and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853-1865, 2017.

[18] J. Ž. I. B. A. P. M. a. B. A. Gama, "A Survey on Concept Drift Adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, 2014.

[19] D. J. C. MacKay, "Bayesian interpolation," *Neural computation*, vol. 4, pp. 415-447, 1992.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 11 2011.