Energy Consumption Prediction Using Data Reduction and Ensemble Learning Techniques

Marsa Thoriq Ahmada[1,*] & Saiful Akbar [1,2]

[1]School of Electrical Engineering and Informatics, Institut Teknologi Bandung,
Jalan Ganesa No. 10 Bandung 40132, Indonesia
[2]University Center of Excellence in Artificial Intelligence for Vision, Natural Language Processing, and Big Data Analytics (U-CoE AI VLB), Institut Teknologi Bandung,
Jalan Ganesa No. 10 Bandung 40132, Indonesia
*E-mail: marsathoriq@gmail.com

Abstract. Building energy problems have various kinds of aspects, one of which is the difficulty of measuring energy efficiency. With current data development, energy efficiency measurements can be made by developing predictive models to estimate future building needs. However, with the massive amount of data, several problems arise regarding data quality and the lack of scalability in terms of computation memory and time in modeling. In this study, we used data reduction and ensemble learning techniques to overcome these problems. We used numerosity reduction, dimension reduction, and a LightGBM model based on boosting added with a bagging technique, which we compared with incremental learning. Our experimental results showed that the numerosity reduction and dimension reduction techniques could speed up the training process and model prediction without reducing the accuracy. Testing the ensemble learning model also revealed that bagging had the best performance in terms of RMSE and speed, with an RMSE of 262.304 and 1.67 times faster than the model with incremental learning.

Keywords: *bagging; boosting; data reduction; dimensionality reduction; energy efficiency; ensemble learning; LightGBM; numerosity reduction.*

1 Introduction

Energy is a field with a wide variety of issues. Energy generation, it can produce emissions of harmful particles such as SO_2, NO_x, and CO_2. These particulate emissions are pollutant materials that are directly related to human health and environmental problems. They can cause diseases of the respiratory tract, global warming, and acid rain. Unfortunately, Indonesia is still dependent on fossil energy. From 2018 data it can be seen that the primary energy sources in Indonesia are dominated by fossil energy such as oil, gas, and coal at around 82.9%, while renewable energy sources such as hydropower and geothermal energy are at 17. 1% [1].

Received December 17th, 2021, 1st Revision August `9th, 2022, 2nd Revision November 2nd, 2022, Accepted for publication November 19th, 2022.
Copyright © 2022 Published by IRCS-ITB, ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2022.16.3.1

One of the efforts to reduce energy consumption is to increase energy efficiency in buildings. Improving energy efficiency in buildings can have a direct impact on reducing the costs incurred. Measuring energy efficiency improvement is not an easy task. This is because there is no sure-fire way to know what the energy consumption was before a building was repaired. Using only a comparison of energy meter values is not enough, because at different times and weather conditions the amount of energy required for the same building will also be different. This measurement error can make fixes that do not actually reduce the efficiency of the building appear to be due to weather and time differences. For example, Rong, *et al.* tried to reduce the energy consumption of a data center [2]. They tried to make improvements by more efficiently using computing and physical resources. But the research did not consider external factors like the weather, which may have affected the result.

These problems could be overcome by developing a predictive model to predict the energy consumption of a building and considering aspects such as time and weather conditions. The prediction results can then be used to see if there is an increase in energy efficiency of the building. With energy measurement sensors and other sensors, data can be obtained that support the prediction of the energy consumption in the building. However, with the large amount of data collected by the sensors, several problems arise, such as the difficulty of monitoring massive data, which causes declining data quality. Another problem that will arise is the data size problem. With large data sizes, a lot of computing resources are used and scalability problems arise. Therefore, we need a way to process the data so that it does not consume high computing resources and is more environmentally friendly.

In overcoming the existing problems, several data mining techniques can be used. The problem of the size of the data can be overcome by reducing the data with numerosity reduction and dimensionality reduction techniques. Meanwhile, overcoming the problem of data accuracy can be done with some appropriate data pre-processing techniques, such as removing noise, overcoming missing values, and removing outliers. In addition, ensemble learning techniques can also be used, i.e., combining several predictive models into a combined model. This technique simulates real life from various points of view so the decision making can take various considerations into account. According to Wan & Yang in [3], this technique is generally able to achieve better performance compared to using a single model. With a slight change in the technique, we hoped it could overcome the problem of the size of the data as well.

2 Related Work

2.1 Data Reduction

Data reduction is a technique that can be used to solve big data problems. This technique aims to reduce data while maintain data quality so as not to lose important information. In this paper, two types of data reduction will be discussed, namely numerosity reduction and dimensionality reduction.

2.1.1 Numerosity Reduction

Numerosity reduction is one way to reduce data by removing some instances from the data. From García Gil, *et al*. [4], the purpose of numerosity reduction is to obtain a subset of the entire data that does not contain redundancy, noise, or less relevant instances, so that the information in the original data is not significantly reduced. Examples of numerosity reduction techniques are random sampling and stratified sampling. Numerosity reduction has the advantage that the computational cost of obtaining the data sample is proportional to the sample size of the data [5]. Other techniques require computational costs that are the same size as for all the data, so numerosity reduction is much better in terms of scalability. Because is does not require high computational resources, it is suitable for use on large datasets.

Random sampling is one of the sampling techniques that can be used for instance reduction. This technique takes several samples at random from an existing dataset with the same probability for each sample. For example, there are N instances in the dataset and n instances are taken randomly. Thus, the probability of an instance being taken is n/N [6]. The sample instance formed from this random sampling will proceed to the next stage. Another sampling technique is stratified sampling. In stratified sampling, the dataset is divided into several strata and then from each stratum a random sample is taken [6]. The strata formed are homogeneous and are usually formed based on the similarity of labels for each instance. With this technique, each stratum is ensured to have an instance that represents it in the formed data sample. This technique is suitable for use in cases of unbalanced data.

2.1.2 Dimensionality Reduction

Dimensionality reduction is a technique for selecting relevant features or summarizing many features into new features. This technique is important to use because data with large dimensions has several problems in terms of computation memory and there is also the curse of dimensionality problem. The curse of dimensionality means that the higher the dimension of the data, the more performance of the predictive model is reduced. Dimensionality reduction techniques that can be used are principal component analysis (PCA) and independent component analysis (ICA).

Principal component analysis (PCA) is a technique used to reduce the dimensions of data by changing the representation of each feature into a particular representation with reduced dimensions [7]. One of the advantages of PCA is that the reduced feature representation is obtained without losing a lot of information. This is because PCA creates a new feature that maximizes the variance of the data. Independent component analysis (ICA) is a dimensionality reduction technique similar to PCA. The difference is that ICA is more focused on making features that are statistically independent, i.e., there is no relationship between the features. This technique is widely used in the case of voice signals where feature reduction cannot be performed with a PCA technique that is only based on feature variance. With ICA, statistically independent voice signals can be separated well. The ICA algorithm aims to find the W matrix, which is an unmixing matrix that can be used to obtain features that are independent from each other [8].

Van Der Maaten, *et al*. [9] compared several dimensionality reduction techniques and showed that PCA is better than several non-linear dimensionality reduction techniques. The study did not compare PCA with ICA, but based on the results of Yau's research mentioned in the related research section, ICA has the potential to excel with certain datasets. Therefore, these two techniques were chosen for the present study.

2.2 Ensemble Learning

The ensemble technique is a technique that can be used to make a model more capable of generalizing by combining several machine learning models that have different approaches from each other. This causes errors made by one data algorithm to be corrected by another algorithm that can properly analyze these conditions. Three ensemble techniques will be discussed is bagging, boosting, and stacking.

2.2.1 Bagging

Bagging, or bootstrap aggregating, is the simplest ensemble techniques compared to the other two. It combines the prediction results of several independent models, after which they are summed and then averaged. This technique uses a bootstrap technique so that each model has different training data from one another. From the total dataset, only N random samples are taken, which are then used in the training process. In this technique there is also another way to combine the results of the models, namely by using a weighted average and giving a weight to each model. An illustration of the bagging algorithm is shown in Figure 1.

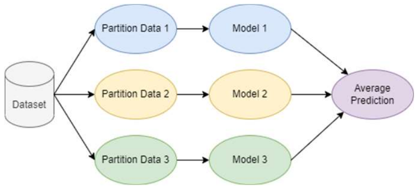

Figure 1 Illustration of the bagging algorithm.

2.2.2 Boosting

Boosting is an ensemble technique that combines several weak models to get a stronger model. A boosting algorithm that is often used is Adaboost, or adaptive boost. In the first step, the Adaboost algorithm trains the dataset with a weak model. Each instance that is incorrectly predicted by the weak model will be given more weight in the next training process. This is repeated until a strong model is formed that can predict better. An illustration of the boosting algorithm is shown in Figure 2.

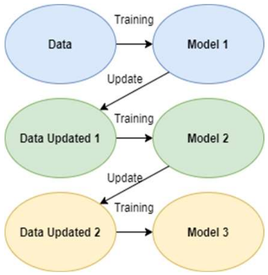

Figure 2 Illustration of the boosting algorithm.

Research related to ensemble learning techniques has been done by Wan and Yang [3]. In their study, several ensemble techniques were compared based on experiments with several datasets. The dataset used was derived from the UCI database, using 31 different datasets. Four ensemble techniques were compared, namely bagging, boosting, stacking, and random forest. According to this paper, the ensemble learning technique with the best performance was boosting combined with bagging. Therefore, the bagging and boosting technique was chosen for the present study.

3 Proposed Solution

3.1 Data Reduction

In the stage after feature engineering, the data reduction stage reduces the size of the data so that it can be processed in a limited computing environment. This stage is divided into two steps, namely numerosity reduction and dimensionality reduction.

3.1.1 Numerosity Reduction

To reduce the data, the first technique applied is numerosity reduction, or instance reduction. In this step, two sampling techniques are used, namely random sampling and stratified sampling. In stratified sampling, the data are divided based on the meter value. Because the meter value is continuous, a clustering technique is used, namely k-means clustering, to make a grouping based on the meter value. From each cluster the same number of samples are taken to represent the cluster. With each technique, four possible amounts of reduced data are tried.

In general, more reduced data will lead to poorer machine learning prediction accuracy but will increase the speed of training and predictions. With a data size of 20 million rows, it is possible for a sufficient large number of data to have redundant information so that if the right number of samples is reduced, the performance will not be degraded and can even increase. By reducing the data, the amount of RAM required to process the data is also reduced. From the various possible value configurations and techniques, the best numerosity reduction technique is selected before proceeding to the next stage.

3.1.2 Dimensionality Reduction

The next stage of data reduction is dimensionality reduction. At first, the features are selected that have to be reduced. The selection of these features aims to avoid reducing important information, instead focusing on reducing features that are less important. The selection of the reduced features is done by looking at the feature importance value from the baseline model. The reduced features are only features with a low feature importance value. After that, PCA and ICA are performed to reduce these features. For each technique, seven possible reduced features are tried. In this stage, also the best configuration from the previous stage is used. The best configuration for the dimensionality reduction technique is taken and evaluated using light gradient boosting before proceeding to the next stage.

3.2 Modeling

After going through all the previous stages, the data is ready to be analyzed by the machine learning model. At this stage, the data is trained by several models and then their performance will be seen. In the selection of the machine learning model, we chose LightGBM Regressor because it has a gradient tree boosting-based approach that is fast and memory-efficient.

Hyperparameter tuning is performed to obtain the best configuration of the model parameters. The hyperparameter tuning technique used is a random search technique, which involves trying many configurations at random so that each model can try various combinations of parameters. The parameters to be tuned in the LightGBM model are boosting type, max depth, learning rate, and n estimators. To overcome RAM problems, two ensemble learning techniques were tested, i.e., bagging and incremental learning. The following is an explanation of the two techniques.

3.2.1 Bagging

Bootstrap aggregating (bagging) is an ensemble technique that combines the prediction results of several independent models, after which the prediction results from each model are added up and then averaged. This technique uses a bootstrap technique so that each model has different training data from the others. From the total dataset, only N random samples are taken, which are used in the training process. This technique is commonly used for applications with high resource use because it combines several models. In addition, the advantage of using several models is that a stable model is obtained that can reduce the problem of prediction variation and avoid the problem of overfitting.

This model can also be used to solve the problem of large data, with minor changes to the bootstrap part. Large data sets require large RAM. During the training stage, RAM usage will increase and errors can occur when there is not enough RAM. By using bagging, the data is divided into several partitions to be trained separately. With a smaller data partition size, less RAM is required so that training on large data sets can be done. In addition, each model will have different errors. It is hoped that by combining the prediction results of each model, the models complement each other.

3.2.2 Incremental Learning

Incremental learning is closely related to learning from streaming data, which happens over time. This is because incremental learning can update the model gradually so that model learning can be done continuously. Incremental learning can be used when a model is no longer relevant to the current data, so retraining is necessary [10]. However, compared to retraining from scratch, it is better to continue training with relevant new data without erasing previous learning. In addition, incremental learning can be done with limited memory resources and, ideally, without compromising the accuracy of the model. Incremental learning can be done with limited memory resources because learning can be done by dividing the data into partitions so that the memory requirements are reduced. Therefore, this technique was considered appropriate for this study, which aimed to process large data with limited memory and computational resources.

4 Evaluation

4.1 Experimental Design

In this section, we will discuss the implementation environment, data collection, and experimental flow in this research.

4.1.1 Implementation Environment

The environment for implementing the solution in a limited environment, namely on Google Collaboratory with 12 GB RAM, 100 GB Memory, Intel Xeon CPU @ 2.00GHz x 2, and a program run time limited to 12 hours by using the Python programming language.

4.1.2 Dataset

The dataset was obtained from the Kaggle platform, namely the ASHRAE - Great Energy Predictor III competition. It contains data on holidays, historical data on energy consumption of a building, data on the building itself, and historical data on the weather in the area of the building. The dataset includes data for a full year in 2016 and hourly energy records. The data came from 1,449 buildings in 16 different locations.

The dataset had about twenty million rows and seventeen columns. With limited implementation resources, the size of the data is quite large. This means that during the training process errors can occur so that the expected model is not obtained. There were also nine features that had missing values.

4.1.3 Experimental Flow

In developing machine learning models, it is necessary to experiment with several parameter configurations. The validation scheme that was used in this stage was Holdout. The evaluation metrics used to measure the prediction accuracy were root mean square error (RMSE) and training time. In this experiment, one factor was used at a time so that only the best configuration in the previous stage was continued in the next stage.

Two main models were tested, i.e., incremental learning and bagging models, and two numerosity reduction techniques were tried, namely random and stratified sampling. Experiments were done with four configurations for each technique. Furthermore, in the dimensionality reduction stage, the PCA and ICA techniques were compared with each of the seven value configurations. Configuring the best model was done by tuning the parameters. In the final stage, the incremental learning and bagging models using the best configuration were compared.

4.2 Data Reduction

The first data reduction used was numerosity reduction. The techniques that were compared in numerosity reduction were random sampling and stratified sampling.

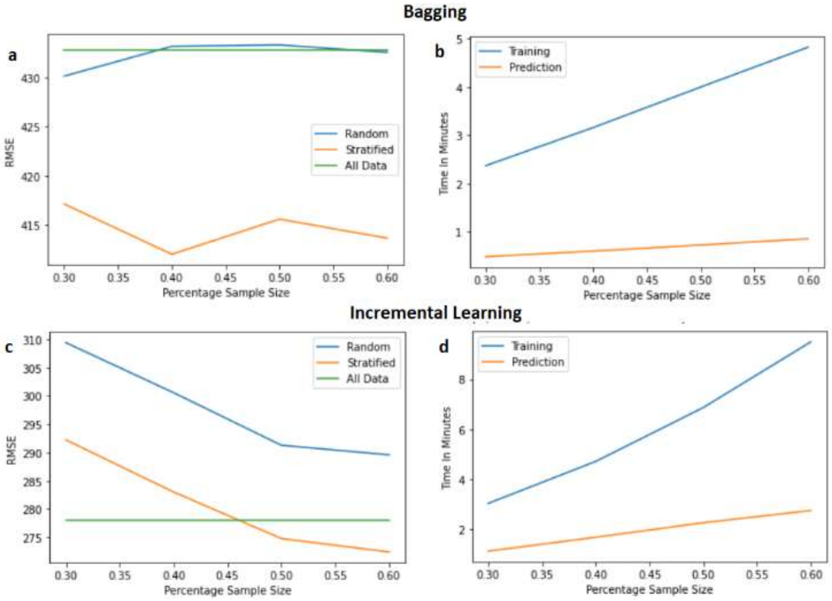

Figure 3 (a) Experimental results of numerosity reduction in the bagging model. (b) Training speed and prediction of numerosity reduction in the bagging model. (c) Experimental results on numerosity reduction in the incremental learning model. (d) Training speed and prediction of numerosity reduction in the incremental learning model.

The training time of the two numerosity reduction techniques tended to be the same because the difference between them only affects the data processing time and not the training time and predictions. Therefore the visualization only displays one. From Figure 3 we can see that for the bagging model, the increase in time was less than for the incremental learning model. This is because in the incremental learning model there is an additional time cost to improve the previous decision tree. The model with the bagging technique performed best with the random sampling technique with a sample size of only 30% of the total data, while the stratified sampling technique used a sample size of 40%. In both techniques, there was an increase in performance compared to using all data. In the model with the incremental learning method, it can be seen that with a larger sample size, the performance of the model tended to be higher. The obtained sample size was 60%, which had the best performance with both sampling techniques. The stratified sampling technique could obtain higher quality data in terms of variance. By doing grouping with the k-means algorithm, samples from each group can be selected in a balanced manner so that the model can learn well.

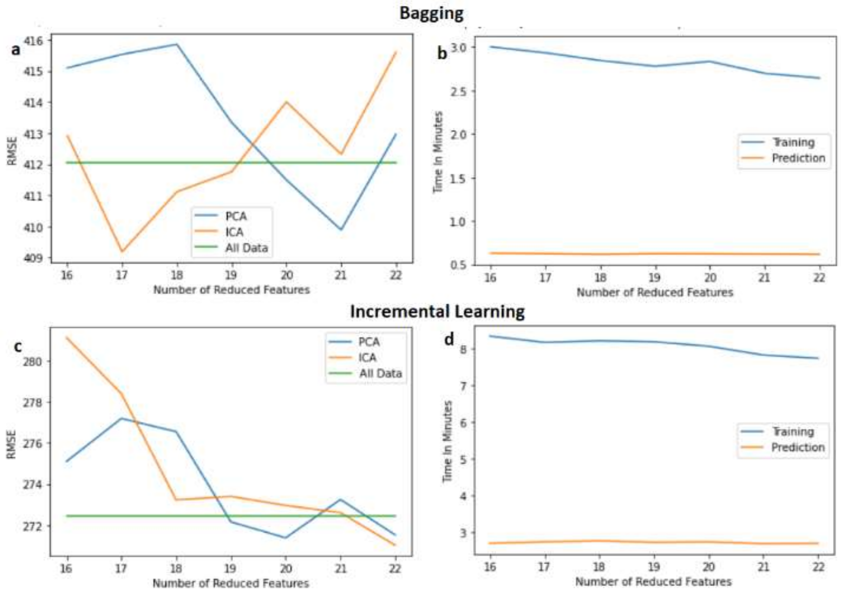

Figure 4 (a) Visualization of the performance of the experimental results for dimensionality reduction with the bagging model. (b) Visualization of training speed and prediction of dimensionality reduction with the bagging model. (c) Visualization of the performance of the experimental results for dimensionality reduction with the incremental learning model. (d) Visualization of training speed and prediction of dimensionality reduction with the incremental learning model.

The second data reduction technique applied was dimensional reduction. The techniques that were compared in dimensional reduction were PCA and ICA. From Figure 4 it can be seen that, in general, the smaller the feature reduction, the higher the training and prediction time. The time used in both models decreased linearly with feature reduction. With both dimensionality reduction techniques, there was no significant difference in the training and prediction times because the use of these techniques only affects the time of data processing.

When viewed in terms of model performance, the bagging model obtained the best performance in terms of the number of reduced features (21) with the PCA technique. Meanwhile, the ICA technique obtained the best performance in terms of the number of reduced features (17). Both of these dimensionality reduction techniques obtained better performance than without the same dimensionality reduction technique. The model that used incremental learning obtained the best performance (in terms of RMSE) with the PCA technique, with a reduction in the number of features by 20. For the ICA technique, the best performance obtained was reducing the number of features by 22. The increase in performance was due to overcoming the curse of dimensionality. Having too many features does not improve model performance and may even worsen machine learning model performance. With a high number of features it is theoretically possible to store more information but in reality it is not optimal because of the possibility of more noise and redundancy in real-world data [11].

4.3 Modeling

In this stage, development and evaluation was done. The models developed in this study were expected to be able to overcome the problem of limited available RAM. By dividing the data into partitions in the bagging and incremental learning models, it is hoped that the training process can done properly without crossing the RAM limit.

In the tuning stage, the best parameters are searched for each model. By optimizing the parameter configuration, it is expected to improve model performance. Tuning was done after all stages, i.e., preprocessing, feature engineering, and data reduction. The best configuration for each stage was used. From the two models that were tuned, tuning was done using a random search method, i.e., looking for random parameter values from the model. In conducting the random search, each model was limited to the same tuning time of twelve hours. The configurations used in this stage were the best configurations from the previous stages. In this test, the parameter configuration with the best performance during the previous tuning result was used.

From the results in the Table 1, it can be concluded that the bagging model performed better than the model with incremental learning. In addition, the bagging model used much less training and prediction time than incremental learning. The performance improvement for the bagging model with parameter tuning was significant because the bagging model could try more parameter configurations than incremental learning within the same tuning time. This is because the training and prediction times of incremental learning are longer, so it can try only a few parameters. After further analysis by calculating the error, namely the difference between the predicted value and the actual value, it was found that the median error value was 15.33, i.e., much smaller than the average error, which was 77.6. With an error value of 15.33, it can be stated that the model performed quite well compared to the median meter reading at 97.125, which means 15.78% of the original value.

Table 1 Result of model comparison experiment.

Model	**MAE**	**RMSE**	**Training Time (in minutes)**	**Predicting Time (in minutes)**
Bagging	**78.014**	**262.304**	9.183	3.286
Incremental learning	82.157	270.562	15.322	6.297

5 Conclusion and Future Work

The bagging model had the best performance in terms of RMSE and speed, with an RMSE of 262.304 and 1.67 times faster than the model with incremental learning. This proves that the bagging method commonly used in data science competitions that consume high computing power can also be used for limited computing resources with some changes in batch data retrieval. This is also proved by the other advantages of the bagging method, namely combining several complementary models that can make the model more stable because it can reduce the variance of the model to avoid the problem of overfitting. In addition, the changes made can create a model that is more scalable in terms of time than the most commonly used method, namely incremental learning.

In the numerosity reduction of the two techniques, namely random and stratified sampling, it was found that the best technique was stratified sampling. This is because stratified sampling can improve the quality of the data from the strata grouping that is made, so that each stratum has data that is balanced with the other strata in the training data. This can overcome problems in the model caused by data inequality, which causes the model to be more biased in predicting the majority value. Meanwhile, for the dimensionality reduction of the two techniques, PCA and ICA, the best results differed depending on the model used. Both techniques improved performance by overcoming the curse of dimensionality problem, so that with both dimensionality reduction techniques the model performance increased and the training speed became faster.

In future work, an effective parallel model for the bagging model can be developed to increase the speed of training and prediction. In addition, other models can be tested that are quite different from LightGBM to see if the techniques mentioned above also work well with these other models.

References

[1] National Energy Council, *Indonesia Energy Outlook 2019*, Jakarta: Secretariat General, National Energy Council, 2019.

[2] Rong, H., Zhang, H., Xiao, S., Li, C. & Hu, C., *Optimizing Energy Consumption for Data Centers*, Renewable and Sustainable Energy Reviews, **58**, pp. 674-691, 2016.

[3] Shaohua, W. & Hua., Y., *Comparison among Methods of Ensemble Learning*, Proceedings - 2013 International Symposium on Biometrics and Security Technologies, ISBAST 2013, pp. 286-290, 2013. DOI: 10.1109/ISBAST.2013.50.

[4] García-Gil, D., Alcalde-Barros, A., Luengo, J., García, S., & Herrera, F., *Big Data Preprocessing as the Bridge between Big Data and Smart Data: BigDaPSpark and BigDaPFlink Libraries*, in IoTBDS, pp. 324-331, 2019. DOI: 10.5220/0007738503240331.

[5] Kalegele, K., Takahashi, H., Sveholm, J., Sasai, K., Kitagata, G. & Kinoshita, T., *Numerosity Reduction for Resource Constrained Learning*, Journal of Information Processing, **21**(2), pp. 329-341, 2013.

[6] Pandey, K. & Shukla, D., *Optimized Sampling Strategy for Big Data Mining through Stratified Sampling*, International Journal of Scientific & Technology Research, **8**(11), pp. 3696-3702, 2019.

[7] Han, J., Kamber, M. & Pei, J., *Data Mining: Concepts and Techniques,* 3rd ed., Amsterdam: Elsevier/Morgan Kaufmann, 2012.

[8] Langlois, D., Chartier, S. & Gosselin, D., *An Introduction to Independent Component Analysis: Infomax and Fastica Algorithms*, Tutorials in Quantitative Methods for Psychology, **6**(1), pp. 31-38, 2010.

[9] Van Der Maaten, L., Postma, E. & Van den Herik, J., *Dimensionality Reduction: A Comparative Review*, Journal of Machine Learning Research, **10**, pp. 66-71, 2009.

[10] Gepperth, A. & Hammer, B., *Incremental Learning Algorithms and Applications*, in: European Symposium On artificial Neural Networks (ESANN), pp. 357-368, 2016.

[11] Verleysen, M., & François, D., *The Curse of Dimensionality in Data Mining and Time Series Prediction*, in: International Work-conference on Artificial Neural Networks, pp. 758-770. Springer, Berlin, Heidelberg, 2005.