Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 7, No. 3, June, 2023

Adaptive Learning and Integrated Use of Information Flow Forecasting Methods

Ilya S. Lebedev [1], Mikhail E. Sukhoparov [2*]

[1] *St. Petersburg Federal Research Centre of the Russian Academy of Sciences, 39, 14th Line V.O., St. Petersburg, 199178, Russia.*

[2] *Russian State Hydrometeorological University (RSHU), 79 Voronezhskaya Ulitsa, St. Petersburg, 192007, Russia.*

Abstract

This research aims to improve quality indicators in solving classification and regression problems based on the adaptive selection of various machine learning models on separate data samples from local segments. The proposed method combines different models and machine learning algorithms on individual subsamples in regression and classification problems based on calculating qualitative indicators and selecting the best models on local sample segments. Detecting data changes and time sequences makes it possible to form samples where the data have different properties (for example, variance, sample fraction, data span, and others). Data segmentation is used to search for trend changes in an algorithm for points in a time series and to provide analytical information. The experiment performance used actual data samples and, as a result, obtained experimental values of the loss function for various classifiers on individual segments and the entire sample. In terms of practical novelty, it is possible to use the obtained results to increase quality indicators in classification and regression problem solutions while developing models and machine learning methods. The proposed method makes it possible to increase classification quality indicators (F-measure, Accuracy, AUC) and forecasting (RMSE) by 1%–8% on average due to segmentation and the assignment of models with the best performance in individual segments.

Keywords:

Machine Learning;
Segmentation;
Adaptive Learning;
Information Local Properties.

Article History:

Received:	30	November	2022
Revised:	14	February	2023
Accepted:	05	March	2023
Available online:	03	May	2023

1- Introduction

Currently, the application of artificial intelligence methods makes it possible, in some cases, to surpass human abilities in solving many problems. Machine learning algorithms make it possible to identify the characteristics, statistical properties, and implicit knowledge necessary to achieve a given result by systematically analyzing sufficient relevant data samples. Machine learning algorithms require the prior extraction of feature values from observable objects to represent input sequences and target variables. Their performance depends to a large extent on the characteristics of the samples analyzed. It is crucial to choose the right group of features that optimally represent the most significant properties of the input data [1, 2]. After that, the model makes it possible to compare the extracted characteristics of the objects with the desired result.

Nevertheless, in many tasks, determining the main characteristics and attributes of the data that will achieve the specified qualitative indicators is a complex and time-consuming process. Simultaneously, in practice, when processing information flows, the concept drift phenomenon occurs when there is a shift in the ranges of target variables and changes in the data distributions. All this leads to the fact that, over time, any model can worsen its qualitative processing performance. Most models are trained on historical data and then used to solve forecasting problems. However, during the functioning of processes, various effects may occur, and the ranges of recorded values may change, which affects the quality of the results obtained by the data processing algorithms.

* **CONTACT**: msukhoparov.rshu@mail.ru

DOI: http://dx.doi.org/10.28991/ESJ-2023-07-03-03

Under these conditions, the data distribution can change over time, which leads to concept drift [1, 3], with changes occurring in the conditional distribution of output data values of input attributes while distributing input data can remain unchanged. Constantly changing data streams characterize processes in various subject areas [4]. A stream processing model should provide specified quality indicators for predicting tasks at high update rates. This necessitates a simultaneous analysis of both the qualitative results of the processing model and the properties of the processed data.

Most machine learning methods use "centralized data", where samples store all the information on the observed objects. Collection processes are performed over some time and usually contain tuples of values when the observed system is in different states and is affected by many heterogeneous factors. This results in phenomena involving the transformation of properties and shifts in the ranges of values obtained from the recording elements. All this leads to the heterogeneity of the data in samples. In separate sequences within a sample, an imbalance of classes, a change of distributions, probabilities of events, and objects of observation can occur.

Machine learning methods can make it difficult to solve prediction problems when various statistical effects occur. For example, if Simpson's paradox [3] appears in the data, the standard approach to centralized intelligent sampling analysis may not achieve the specified qualitative indicators of data processing, and the processing result may not correspond to the true state. Modern approaches to building processing models involve forming, analyzing, and combining local results, which use aggregation methods. The methods and algorithms that solve the classification and regression problems may have different results for the selected quality indicator on the same data set. The values of different classifiers obtained in processing objects of observation can differ. They are considered complementary. By integrating several models, it is possible to improve the quality of classification in some cases.

Currently, ensemble methods are dominant. Among them, the most known are approaches based on simple, combined voting [4, 5] and the application of several aggregating functions that calculate the maximum, average, median, and other class probabilities, averaging the prediction result on a set of responses. Alternatively, various aggregators based on ranking classification algorithms, arbitrators, and combinators [5, 6] are used that apply to both binary and multiclass problems. Another direction relates to the formation of samples. Some researchers (e.g., [4, 6–10]) investigated various aspects of vertically distributed data and proposed technologies, basic algorithms, and combined strategies to select observation objects, allowing us to obtain the main characteristics of sequences and samples and exclude from consideration the values that lead to distortion of data properties [11–14].

Recently, some fields have used hybrid classifiers. Combinations of methods, where different models are based on relatively simple classification algorithms and complex neural networks, achieve high rates of completeness and accuracy [14, 15]. However, the capabilities of a single model depend on the properties of the training sample, and if the characteristics of the data change, the quality indicators can decrease significantly [16–20]. The accuracy and completeness of processing results depend on many factors [21, 22]. The application of such approaches often leads to various situations where the aggregation of different models not only does not help improve the quality indicators but, in contrast, worsens the results [23–25]. Such effects are often leveled on a large sample of data but are clearly visible in its segments. This leads to the fact that errors in the processing of data streams are possible due to different settings of the classification models [26–28].

Thus, it is necessary to develop new strategies and adapt existing ones that enable accurate and reliable training within separating functions and samples. Almost all proposed approaches, methods, and algorithms for machine learning today are highly specialized [29–32]. Each model achieves particular qualitative indicators for those subject areas where it was optimized and for the data on which it was trained. One of the main problems in achieving qualitative indicators in machine learning methods is related to the fact that when the properties of incoming data change, a need for additional training occurs [33–36]. Most models that solve prediction problems are trained on a predetermined set of observational objects. In the case of transformations in the properties of information sequences, the quality of processing decreases. Thus, there is a need to improve the completeness and accuracy of model classification in prediction problems under the influence of external factors.

The approach proposed in this paper is based on partitioning data samples into subsamples with their own properties. These properties allow us to choose the most efficient algorithms and models for classification tasks and the prediction of time series. The novelty of the proposed method is that the sample is pre-partitioned into subsamples based on the calculated information about the variations in the ranges of the target variables and predictors. The use of models to detect concept drift allows the real-time formation of subspaces of data with their properties, which can be used in the future for continuous learning and monitoring of the performance of the models. This study improves the quality performance of the prediction problem based on the segmentation and adaptive selection of different machine learning models on selected segments of the local data sample.

The rest of the paper is organized as follows: Section 2 describes the formalized problem statement and the method developed in this study. Section 3 presents the test results based on the experiments performed. Section 4 discusses the applicability of the approach considered in this study. The conclusion is an interpretation of the results.

2- Proposed Method

2-1- Basic Notation

The use of models whose improvements are based on updated local information is one of the problematic issues in classification and regression [18]. Typically, the training sample is considered a single set. However, the data tuples comprising these can be obtained under the influence of various factors. For example, the appearance of individual control commands increases the number of service messages in the network traffic. The change of seasons and the increasing length of the day are reflected in the power consumption of the power supply systems. Many factors that affect the values of the training set variables are known in advance. In this regard, it becomes possible to identify the training sample tuples received at the time of exposure.

Let X^q be a sample with a size of$q$, $\{a_1,\ a_n,\dots a_r\} \in A$be the set of basic classification algorithms. The problem arises of determining a the classification algorithm that is most suitable for data sampling of a given quality indicator.

A set of factors is affecting $\{v_1,\ v_2,\dots v_m\} \in V$ the values of target variables in$x \in X^q$ tuples.

$L(a, x, v)$ is the loss function at the time of factor v.

The quality functional is determined by an expression related to the action of the factor *v;*

$$Q(a, X^q, V) = \frac{1}{q}\sum_{x \in X} L(a, x, v) \quad (1)$$

Thus, it is then necessary to minimize the functional:

$$Q(a, X^q, V) \to min, \quad (2)$$

which makes it possible to assign algorithms $a \in A$for a sample, X^q during the formation of which the ranges of variable values were influenced by factors V. Such a formulation allows for consideration of the influence of known factors that can cause effects affecting the spread, the bias of the classifiers' answers.

2-2- Method Description

One of the problematic issues with adapting machine learning models is the lack of effective methods of information pre-processing aimed at calculating and analyzing properties that allow dividing incoming sequences into segments in real time. Such complex methods should solve not only the usual problems of filtering, noise removal, and emissions but also provide information about the properties of the data to select and determine the most suitable models. Figure 1 shows an example of the model.

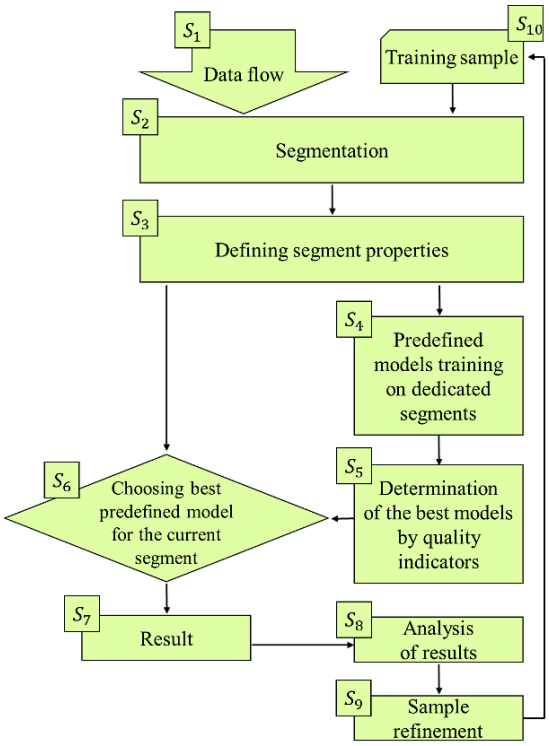


Figure 1. Flowchart of method steps

The model shown in Figure 1 has two parts. In the left part, a continuous information flow is processed; in the right part, procedures ensure the implementation of the "mechanism" of training and the selection of the most effective model that solves problems of classification, regression, and prediction. A feature of the presented solution is the segmentation of the data sample, which allows for the preliminary pre-training and tuning of algorithms. Let us consider many steps to implement the method.

S_1. For the initial start of processes, it is necessary to have preliminary information about the values of $x_1, \dots\ x_n$ the information sequence. They were included in the initial training set.

S_2. The sample is analyzed to determine individual segments where data properties differ. Its separation is possible both on the basis of a predetermined system of rules, and with the help of algorithms that automatically search for characteristic points where the properties of incoming information sequences change. The separation of objects of observation can be carried out using models, methods, algorithms that calculate the points of decomposition, the change of concept. They automatically define the segment boundaries.

S_3. The initial sample was divided into several parts $X_1, \dots X_m$. Their properties $H_1, \dots H_m$ were analyzed. Depending on the algorithm underlying segmentation, it is possible to determine the direction of the trend, the probability density of the analyzed events, etc. The properties and characteristics of the segments are analyzed, and if there is a match, it is possible to reduce the number of segments under consideration.

S_4. Subsamples $X_1, \dots X_m$are received with the input of models. Their $a_1,\ a, \dots a_r$ training and analysis of the achieved quality indicators occur.

S_5. On each segment X_i the quality functional $\mathrm{Q}(a_i(x), X_i)$ is determined for each model$a_i(x)$. Based on its values, it is possible to rank models $\{a_1, \dots ,\ a_r\} \in A$and select those with the highest quality indicators for each segment.

S_6. In parallel with the right part, the procedures for segmenting and determining the properties of the data sequence are performed when processing incoming data. Analyzing the properties of the segments identified during the processing of the information flow and comparing them with the properties of the subsamples obtained from the training sample allows you to assign one of the pre-trained models $\{a_1, \dots ,\ a_r\} \in A$to the current segment. The selected $a_i(x)$model is used to solve flow processing problems.

S_7. The selected model gives the result of processing.

S_8. The obtained results are compared with the available ones, their qualitative indicators are analyzed.

S_9. Comparison of the obtained model and real values allows you to decide on the formation of data to refine the algorithm, which is subsequently added to the training sample.

S_{10}. Gathering information for updating the training sample.

Thus, it is possible to implement a constantly learning method, where the processes of learning and processing information flows can be carried out in parallel. In the case of using complex classification or regression models, pre-trained models can reduce the time spent on training when the data properties change.

Currently, no single algorithm works well with all data. It is difficult to predict which learning algorithm is appropriate for a particular set in advance. In this regard, the problem of combining several classifiers into a single structure to obtain a better decision-making model arises. The effectiveness of employing various classifiers depends on the information properties and features of their algorithms. Algorithms with different characteristics are selected depending on the tasks to be solved and the required characteristics of speed, completeness, and accuracy. Each learning algorithm uses its own methodology for the dataset. Some of them require "fine-tuning"; others have a high processing speed, and still others are sensitive or insensitive to outliers.

The discussed approach to processing dynamic information flows proposes to aggregate machine-learning algorithms tuned to data properties.

The information flow, represented by the information data sequence, is sent for pre-processing. Data properties are evaluated, and the sample is split into separate segments with matching data properties. As a result, the X^q set is split into subsets, $X_1,\ X_2, \dots X_m$ each having properties different from the others.

Pre-processing allows for adaptive tuning of the base classifiers $\{a_1,\ a_2, \dots a_r\} \in A$ on each $X_i \in X^q$ subset. They are trained to form the parameters and weight matrices of classification algorithms, and then their results are analyzed. The selection is made based on minimizing the empirical risk aimed at finding an algorithm for which, at $x_i \in X_i$the following condition is satisfied:

$$a(x) = \underset{a_i \in \mathrm{A}}{arg\min}\, \mathrm{Q}(a_i(x_i), X_i), \tag{3}$$

Using Equation 3 for each subsample, it becomes possible to choose the algorithm with the best performance indicators. A simple aggregate function, calculating the best algorithms on $X_1,\ X_2, \ldots X_m$ subsamples for Equation 3, will take the form:

$$a(\mathrm{x}) = F(\underset{a_i \in A}{arg\min}(\mathrm{Q}(a_1(x_1), X_1), \ldots, a_r(x_1), X_1)), \ldots, \underset{a_i \in A}{arg\min}(\mathrm{Q}(a_1(x_m), X_m), \ldots, a_r(x_m), X_m))) \quad (4)$$

Depending on the problem being solved, the preset qualitative indicators, the data properties, and the features of the training subsets, it is possible to form more complex functions that consider the classifiers' weights and use additive coefficients, changing the "importance of the voice" of the algorithm. The obtained processing result is then fed into the training sample and regarded by the preprocessing algorithm to refine the model. The proposed multilevel approach evaluates possible algorithms at the preliminary stage, the subsequent selection of the algorithm, and its aggregation with others. While implementing complex machine learning models, several problematic issues arise related to the effectiveness of applying its individual components. Each basic algorithm has different performance indicators for data with different properties. In real systems, the frequency of the observed events may change, the range of values may shift, and an imbalance may appear in the dataset over time. In this regard, it is necessary to develop effective methods for information pre-processing to calculate and analyze the properties entering the analyzer input. They must perform the usual tasks of filtering, removing noise and outliers, calculating the data properties, and forming their segments. A set of such methods should be used to select and determine the most appropriate models for classification and regression problems.

The application of these methods is based on pre-processing, where individual segments with similar properties are differentiated from the initial sample. For example, in regression problems, these can be trends and seasonal changes. With the automatic separation method, points are calculated at which the direction of the trend changes or situations of concept change are analyzed.

The proposed method initially assumes that known factors affect the data properties. These can be commands, control actions, or events associated with a change in the environment. The formed training sample consisted of tuples. Their values are obtained under the action of these factors. Information about internal and external influences is used to divide subsets in such a way as to reduce the number of noise objects and improve the class distinguish ability properties. The application of the method can be considered for classification problems. Let us consider the error indicator I as a function for measuring the losses of the classification algorithm $Q(a_i(x), X_i)$ is determined for each model $a(x_i)$acting on the X^qsample.

$$I(x, a) = \left(x_i \neq a(x_i)\right) \quad (5)$$

The error rate υof the $a(x)$algorithm is determined by the following expression:

$$\upsilon(a, X^q) = \frac{1}{q}\sum_{i=1}^{q} I(x_i, a(x_i)) \quad (6)$$

The recorded data are affected by factors V. Factorscan be defined explicitly; for example, for many datasets in the field of power generation, it can be seen that the length of daylight hours, working, and non-working hours may significantly affect power consumption. However, it is sometimes impossible to unambiguously interpret their effects due to their large number and complicated interpretation. To improve the performance indicators of machine learning methods affected by data outliers, noise, or changes in the density of the probability of occurrence, it is necessary to split the X^pset into subsets regarding the influence of factors $v_i \in Vi = 1, \ldots, m$ on the data m. In essence, the impact of factors is expressed as a change in the ranges of target variables. Such moments are tracked by various methods to detect concept drift. The set can be split by analyzing the data properties in the information flow, for example, the density of the probability of occurrence of the classified events. Various methods are used for this. Some of the simplest are DDM and SEED.

The Drift Detection Method (DDM) [37] uses the binomial distribution, which represents the probability of a random number of errors in a sample consisting of n examples. For each j-th object of observation from the X^q set, the probability of misclassification p_j with standard deviation is $s_j = \sqrt{\frac{p_j(1-p_j)}{j}}$.

It is assumed (PAC learning model) that with the increasing number of examples, the error rate of the learning algorithm p_j will decrease if the distribution of examples remains stationary. A significant increase in the error rate indicates that the class distribution has changed and, therefore, will change the properties of the class distributions.

The DDM method considers the concept change ratios $p_j + s_j > p_{min} + 2s_{min}$ for the warning level. Beyond this level, the context change is possible$p_j + s_j \geq p_{min} + 3s_{min}$for the drift level. Beyond this level, the concept drift is assumed to be correct, the model caused by the training method is reset, and the new model is trained using the examples saved since the warning level was fired. The values for $p_{min}\ s_{min}$ are also reset.

Huang et al. [38] proposed an algorithm in which blocks of a fixed size are formed for the data. In this regard, by controlling the initial settings of the blocks for training samples, it is possible to form the number of candidate points for changing the concept. By determining the start and end points of a block, neighboring blocks are calculated and examined, and then, if the statistical properties match, they are grouped together. This operation, called "block compression," removes possible change points that are less probable to be true change points. SEED compares two sub-windows. When the two windows have different mean values, the old sub-window is discarded. The SEED parameters in MOA are the block size, the compression ratio, and the threshold, a parameter that controls the size of the increment.

The concept drift definition points make it possible to determine the splitting of the X^q set into non-intersecting sets: $X^q = X_1^{q_1} \cup X_2^{q_2} \cup \ldots \cup X_m^{qm} \sum_{i=1}^{m} q_i = q$, where q is the sample size, q_i is the I segment size.

Each $X_i^{q_i} \in X^q$ subset formed because of analyzing the action of factors for training the classifying algorithm can be split into the training and control samples, $X_i^{q_i} = X_{n_i}^{l_i} \cup X_{n_i}^{k_i} q_i = l_i + k_i$ where $n_i = 1, \ldots, N_i$ (N_i – total number of objects in the segment) are splitting options for $X_i^{q_i}$ sample, l_i and k_i are the lengths of the training and control segment subsamples.

The empirical risk functional Q_i for the $X_i^{q_i}$ sample determined by the influence of factor v_i is:

$$Q_i\left(a, X_i^{q_i}\right) = \frac{1}{N_i} \sum_{ni=1}^{N_i} \upsilon(a(X_{n_i}^{l_i}), X_{n_i}^{k_i}), \tag{7}$$

where υ is the error frequency determined in Equation 6.

The subset obtained with the regard to the influence of factors can be assigned a classifier.

By assuming that the sample is simple and repeats the properties of the general population, it becomes possible to consider the algorithm versions $a = \{a_1, a_2, \ldots a_r\}$ and choose the classifier a_j for the $X_i^{q_i}$ set subject to the condition:

$$Q_i\left(a, X_i^{q_i}\right) = \underset{a_j \in a}{arg\min}\, Q_i\left(a_j, X_i^{q_i}\right) \tag{8}$$

Here, the classifying algorithm can be trained separately on each data segment. Due to the manipulation of samples, it is possible to improve the performance indicators of algorithms in some cases.

The input tuples $x = (x_1, \ldots, x_w)$ of $x \in X$ valuesare detected concerning $x^{q_i} = \left(x_1^{q_i}, \ldots, x_\omega^{q_i}\right)$ values of the resulting disjoint subsets $x^{q_i} \in X_i^{q_i}$, using evaluation methods (for example, KNN, DTW-KNN, neural networks, etc.). In a simpler version, a distance metric can be applied, where the Euclidean distance is used to measure proximity, $d(x, x^{q_i}) = \sqrt{\sum_{t=1}^{\omega}\left(x_t - x_t^{q_i}\right)^2}$ where ω is the window length.

The obtained processing result is analyzed and can participate in forming the $X_i^{q_i}$ sample. Later, preference is given to the most appropriate model trained on the subset selected using the proximity measure.

Model training is complicated not only by the large dimension of the attribute space but also by the presence of variable factors influencing the values of the attributes.

The main limitation of machine learning methods is that classification algorithms cannot always be effective in a system constantly functioning under the influence of various external and internal actions. The system is dynamic; there are constant transitions from one state to another. External and internal factors change the values of characteristics [39–44].

The analysis and consideration of the factors influencing these data make it possible to split the set into subsets. In the future, by determining the properties of the obtained samples, it will be possible to solve the problem of applying the most efficient processing algorithms.

A general view of the processing algorithm is shown in Figure 2.

Algorithm 1: Information sequence processing algorithm

```
Input: flow sample x, dataset X, methods {a₁,…,aᵣ}∈A, split method
Output: update X, method for flow sample a(x)
While x
Separate by split method from X segments X₁...Xₘ
foreach segment Xi do
foreach method aⱼdo
Train aⱼ(xᵢ,Xᵢ)
end
```

Calculate$a(x_i) \leftarrow \underset{a_j \in A}{arg\min}\, Q(a_j(x_i), X_i)$
S←[a_j,X_i]
end

foreach segment $X_i$do
Calculate H←Compare_specifications (x,X_i)
end
j ← min(H)
a(x) ← a_j(x)
X←x
end

Figure 2. The general view of the processing algorithm

2-3-Application Method for Single Classifiers

Consider the classifier $a(x,\ W)$. Tuple *x* arrives at the input. The parameter matrix of the trained classifier is used for decision-making. Two ways of splitting datasets are possible: production rules and membership functions. The use of production assumes that the factors influencing the data values are computable. It is possible to form various subsets from the incoming information flow by analyzing changes in data properties. On the formed training sample, it is possible to determine changes in the properties of the sample data and densities of the probability of occurrence of the events under study for the data, which can be done by determining the points of concept change. The data and their properties that consider the effects on the sample values should be calculated. Generally, such a model is presented in predictive form:

$$\Omega = < A, W, X > \tag{9}$$

Denotes classifying algorithms that use weight matrices to compare the incoming data vector. W is the set of parameter matrices of trained classifiers. The matrix values depend on the factors that affect the data in the system. X is a set of object descriptions consisting of a subset of data samples. Each subset has its own classifier weight matrix.

Values of $w_j \in W$ may be selected on the basis of the production model. The $X_j$data subset is determined concerning the influencing factor. Each subset can be assigned a classifying algorithm $a \in A$.Grouped variable subset $X_i$determines the weight matrix $w_j$regarding the properties of the classifying algorithms. This allows using w_j matrix on the sample determined by the concept change detector. The new tuple *x* arriving at the input is identified by the classifier $a(x, w_i)$.

The segmentation of the dataset will consider changes and improve the performance indicators of the classification model as a whole. The other direction is based on the use of the membership function. It can be used when there are impacts that can be analytically described (for example, the seasonal length of the daylight hours, the latitude of the place in the subsystems for supplying electricity to urban facilities, peak load hours in the information system). Let V be the set of factors influencing the target variables in the data sample.

Such factors $v_j \in V$can be processed using the membership function (indicator function). Based on this, the data sample X is split into a finite number of non-intersecting measurable subsets $X_1 \cup X_2 \cup ... \cup X_m$. In the simplest case, the membership function μ of the subset $X_i \in X$, where x is a training sample tuple, can be represented as:

$$\mu_{X_i}(x) = \begin{cases} 1, x \in X_i \\ 0, x \notin X_i \end{cases} \tag{10}$$

Equation10 makes it possible to determine the membership of an element of the $x \in X$data sample in the $X_i$subset at the time of factor v_i action.

In the general case, the sample consists of $m$subsets. Membership of the subset is determined by the functions $\mu_{X_i}(x)$. Classification for $c_i \in C$class becomes possible on each subset $X_1 \cup X_2 \cup ... \cup X_m$. Test and training samples are formed concerning the acting factors v_j. The classifier $a(x,\ w)$can be supplemented with the $\psi(v_j)$ function depending on the subset being processed. The $\psi(v_i)$ function considers the factor, v_i influencing theX_i, subset and determines the weight matrix $w_i = \psi(v_i)$by its value. The classifier takes the form of $a(x,\ \psi(v_i))$.

The loss function can be used as one of the model evaluation measures for regression problems. The loss function $L(v_j)$ for the X_j subset is determined by the expression:

$$L(v_j) = \frac{1}{N}\sum_{i=1}^{N}(L_i(\varphi(x_i), \psi(v_j)), c_i) \tag{11}$$

where N is the number of observation objects in the subset.

The average amount of losses for the data of the X set is:

$$L = \frac{1}{M}\sum_{j=1}^{M} L(v_j) \quad (12)$$

where M is the number sample affecting factors.

Applying Equations 11 and 12 and minimizing, it is possible to search for optimal parameters based on the expression:

$$L = \frac{1}{MN}\sum_{j=1}^{M}\sum_{i=1}^{N}(L_i(\varphi(x_i, \psi(v_j)), c_i) \rightarrow min \quad (13)$$

Equation 13 makes it possible to determine the qualitative indicator of the classifier loss function, considering the splitting of the data sample.

3- Evaluation

3-1- Experimental Setting

To confirm the advantages of the proposed approach, four publicly available datasets were used. They contain information sequences of electricity generation data. Time series data from conventional power plants and renewable energy sources were used for modeling [45–48]. The proposed solution is based on sample segmentation. A data sequence is being processed. The sample is analyzed, and individual segments are determined. In the experiment, its selection is performed both based on a predetermined system of rules and with the help of algorithms that automatically search for characteristic points where the incoming data properties change. In the case of using heuristics, the sequence is studied. Based on the analysis of the dataset, trends, periods, segments, and clusters with different characteristics are distinguished. Observation object separation can be carried out using models, methods, or algorithms that calculate the points of decomposition, a change of concept. The algorithm for the selected processing model is presented in Figure 3.

Algorithm 2: Determination of indicators of the processing model algorithm

Input: dataset X, methods $\{a_1,\ldots,a_r\}\in A$, split *method*
Output: structure S = (method, segment)
    Separate *by split method* from X segments $X_1 \ldots X_m$
foreach segment Xi do
foreach method a_j do
        Train $a_j(x, X_i)$
end
    Calculate $a(x) \leftarrow \underset{a_j \in A}{arg\min}\, Q(a_j(x), X_i)$
$S \leftarrow [a_j, X_i]$
end
return S

Figure 3. Algorithm for the selected processing model

3-2- Data Processing

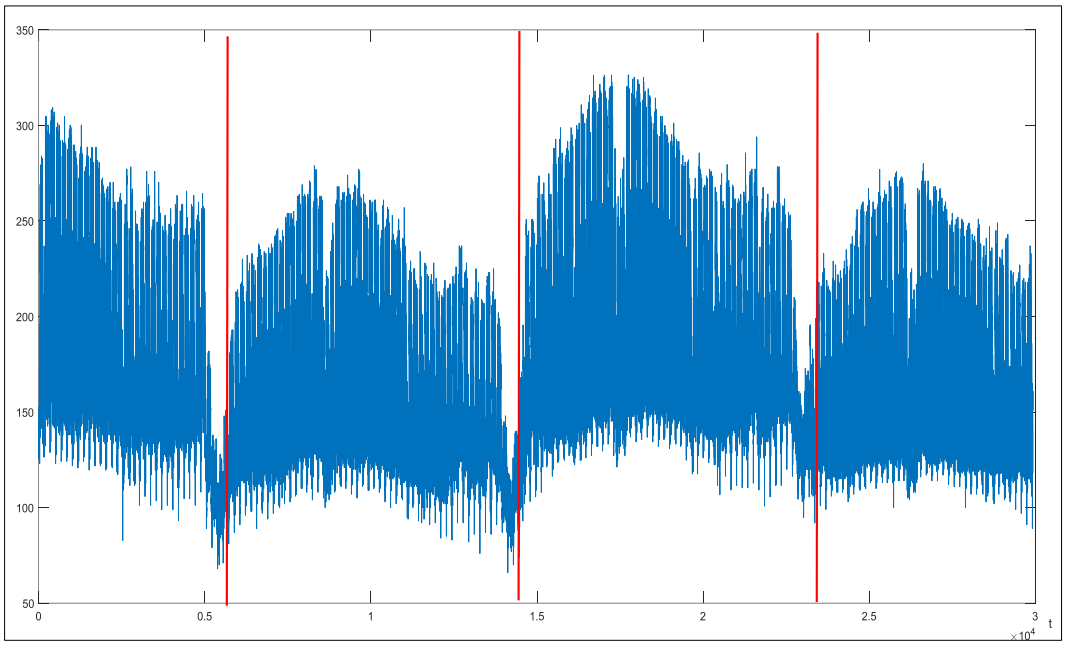
Several datasets were considered experimental data [45–48], which contained data on electricity generation in various regions from 1995 to 2020. Classification and regression problems were considered. The classification quality indicators (AUC – area under the ROC curve, accuracy, and F-measure) and forecasting (RMSE) were evaluated for all samples entirely and using segmentation.

The Power Supply dataset was chosen as the first experimental data; it shows the power supply capacities of the two stations [45]. The choice of the dataset was justified by its structure, containing two predictors, and the ability to determine periods based on the timestamps of records. Two classes were determined by the values of two predictors: working hours and non-working hours. The experiment considered the problem of determining working and non-working hours according to the readings of the capacities supplied to the municipal network from two substations. The seasonal effect was determined to be an influencing factor.

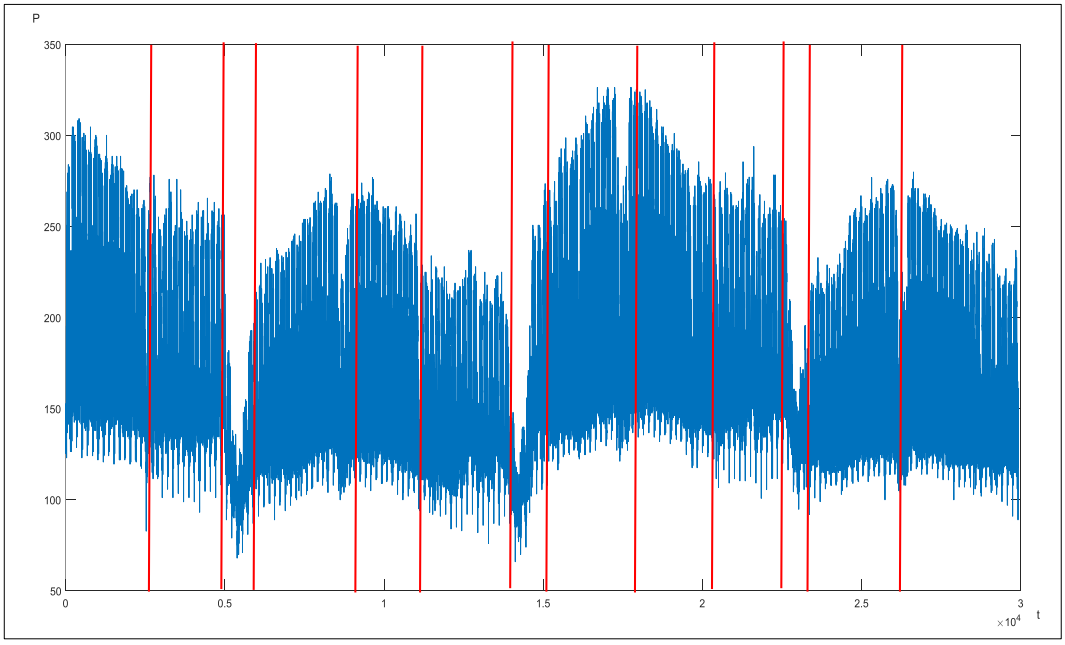
Figure 4 shows a general view of the data. The axes show the days of observations and hours in the horizontal plane, and the power consumption is plotted along the vertical axis.

The concept detector is used to select several points where the density of probability of occurrence changes. The points determined by the detector make it possible to determine the dataset split into subsets, where their properties change. Simultaneously, it is possible to identify segments based on a heuristic approach using membership functions.

At the beginning (Figure 4-a), when analyzing the dataset using the SEED method, four segments were obtained by selecting parameters; these segments were compared to the segments determined by the membership function describing seasonality. Then the window size was reduced, which increased the number of identifiable concept change points (Figure 4-b).



(a)



(b)

Figure 4. Segmented time series of the SEED electricity consumption dataset with different window widths

During the experiment, segments were first obtained using SEED. Then, their analysis was followed by the comparison of the segments determined on the basis of a time scale showing the calendar change of seasons. In the experiment, similar segments obtained automatically using the SEED method and calculated by the membership function were compared to analyze performance indicators. Consider the resulting subsets obtained based on SEED and membership functions.

On Figure 5, the axes show the values of the powers generated by the two power plants. Seasonal factors are used to segment data based on rules defined by the membership function and the SEED method. Figure 5 shows the values of the general population of the entire sample of light areas of working days, shaded areas of non-working days in the winter period.

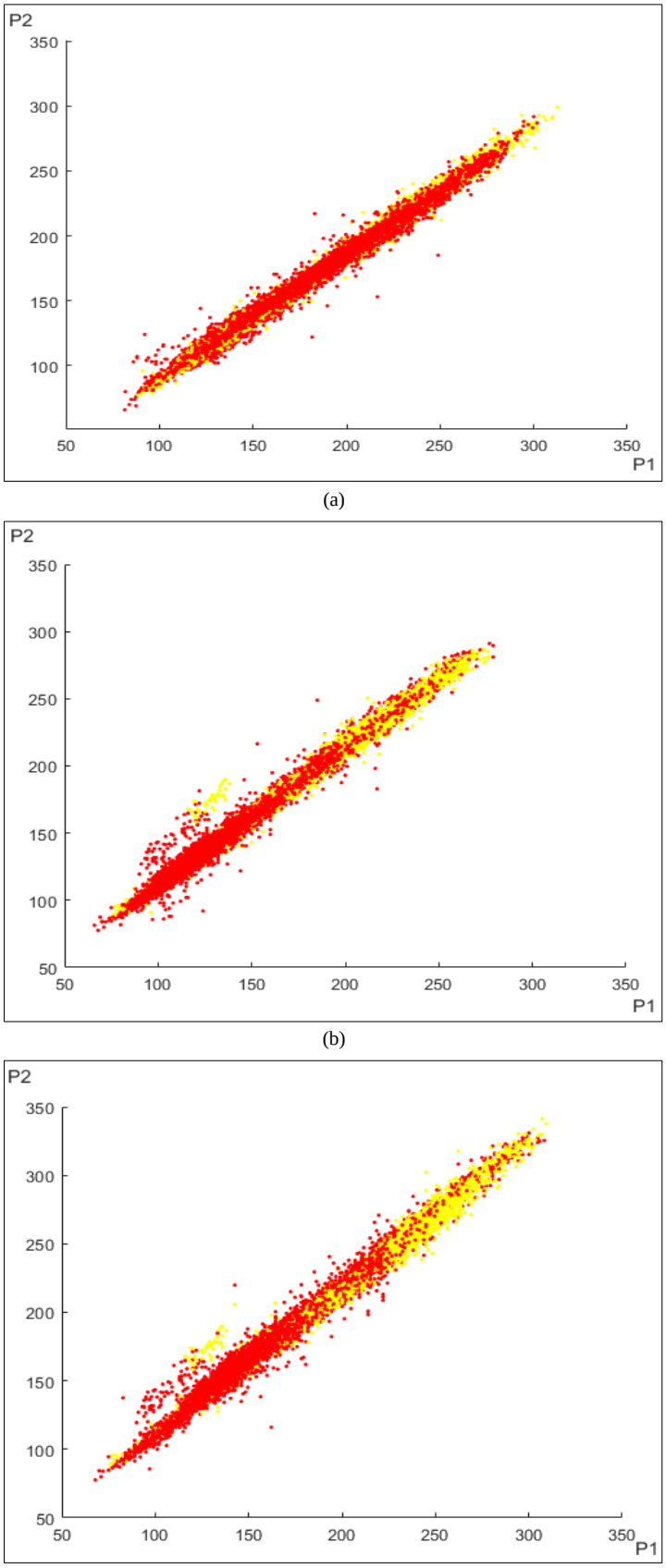
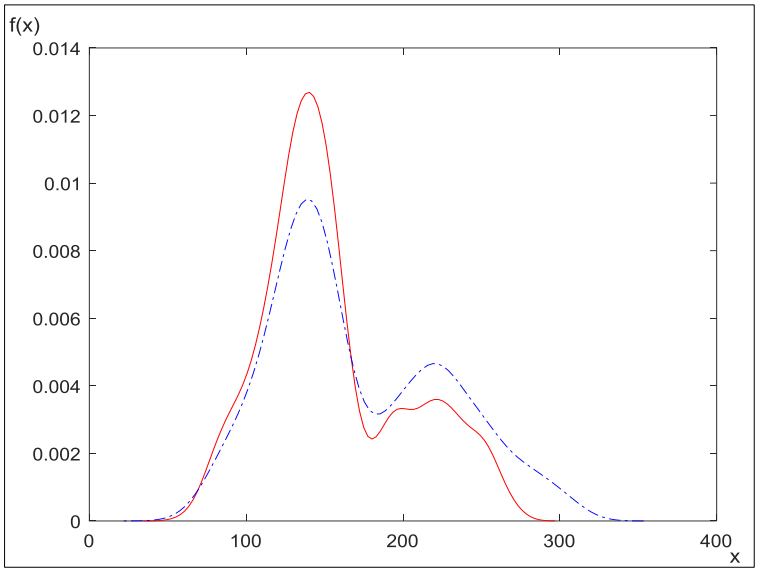
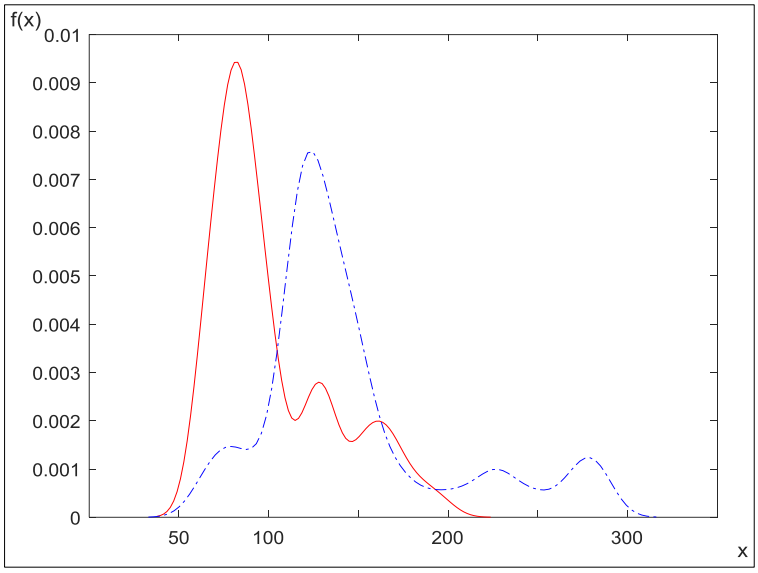


Figure 5. A subset of working and non-working day's power generation in four-part segmentation: a) full dataset, b) winter segment after segmentation by the membership function, c) «conditional» winter segment after segmentation by the SEED method

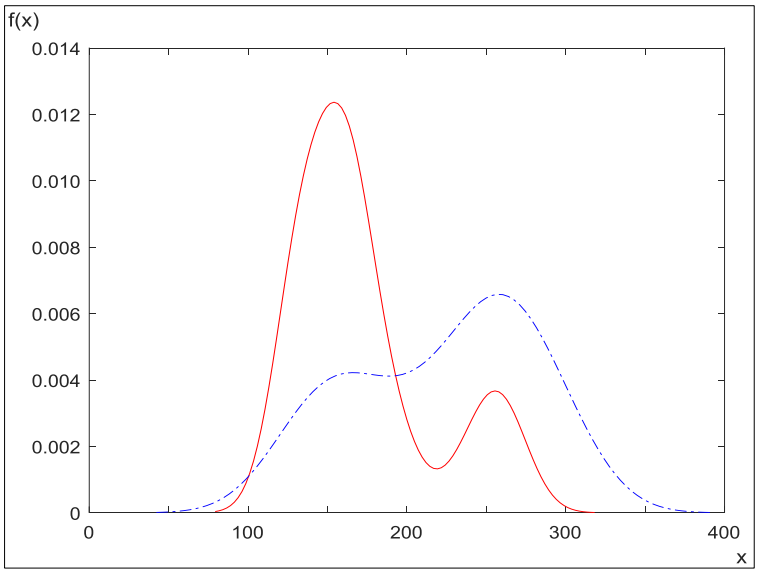
Compared to the data of the entire set, there is a shift in the ranges of variables. Using the information about individual factors that affect the values, reducing the range of data change by segmenting the sample becomes possible. In Figure 6, based on the frequencies of values, a probability density estimation function for working (blue) and non-working (red) time is built for the SEED method and membership function.



(a)



(b)



(c)

Figure 6. An example of a probability density estimation function for classes during working and non-working days power generation: a) full dataset, b) winter segment after segmentation by the membership function, c) «conditional» winter segment after segmentation by the SEED method.

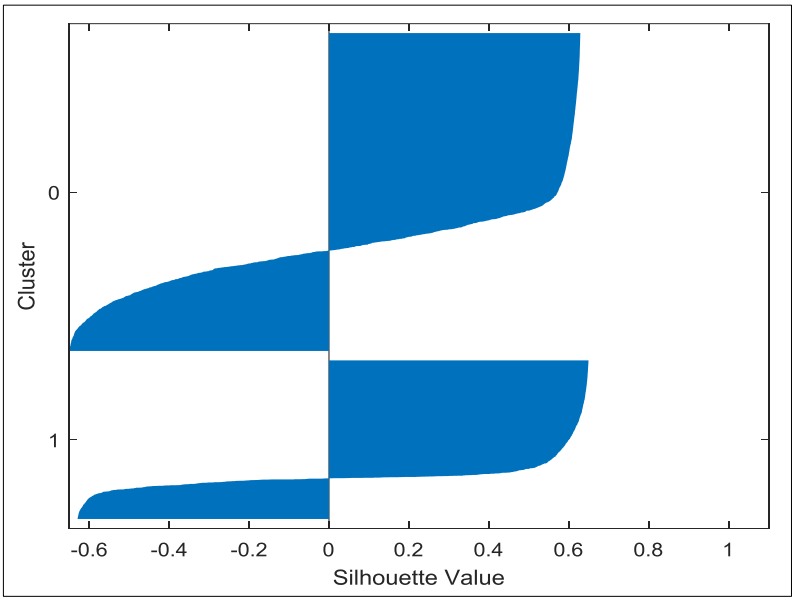
In the case of superposition of graphs, overlapping areas make it possible to estimate the probabilities of occurrence of errors of the first and second kind. The intersection area of the entire set data is larger than that for a subset segment of the winter months. Subsets can be estimated using the silhouette coefficient. The compactness hypothesis specifies that sequences belonging to one target class will be close to each other and far from an object of another class. It is assumed that data values of the same class are grouped side by side. Therefore, the silhouette function is used to evaluate such clusters. It is possible to check the consistency of data in areas with its help.

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{14}$$

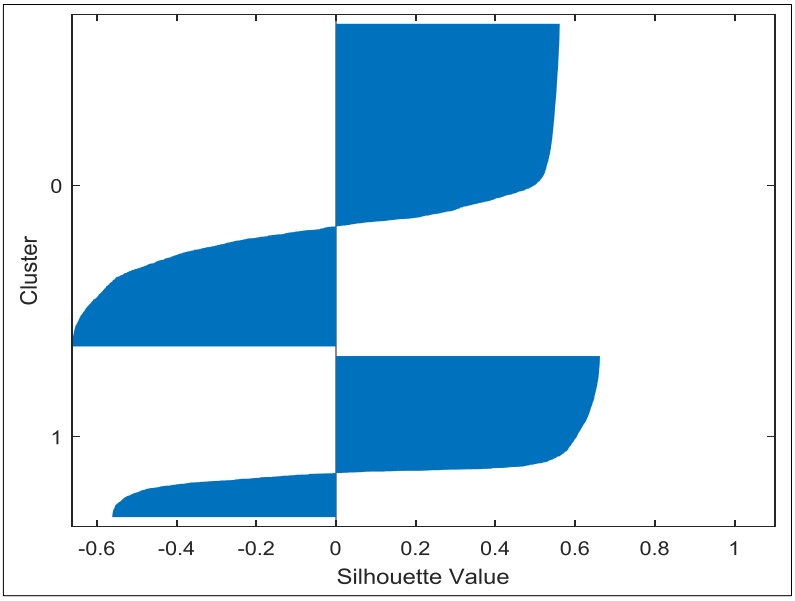
where a_i is the average distance from the i-th point to other points in the same cluster as, i and b_i is the minimum average distance from the i-th point to points in the other cluster. The entire cluster structure was evaluated as an average of the indicators for the elements:

$$SWC = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{15}$$

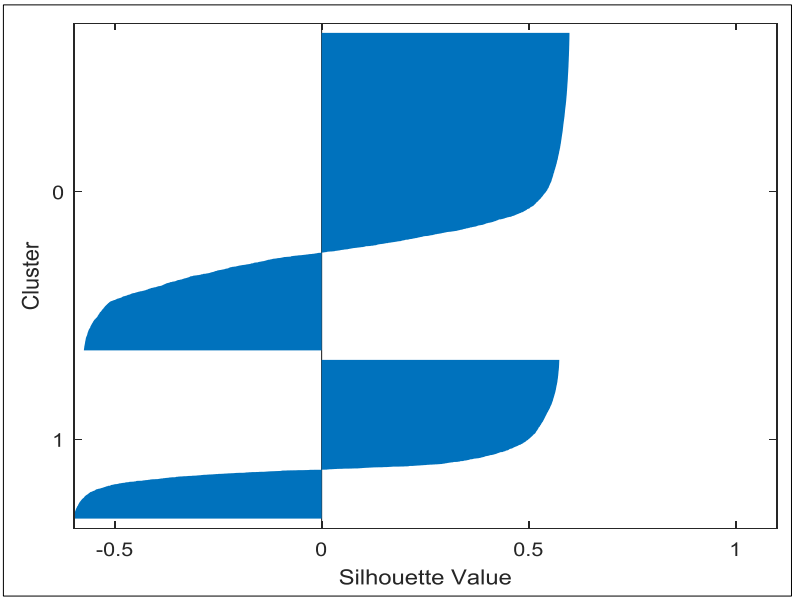
Figure 7 shows the graphs of the silhouette coefficient for the segments obtained automatically and based on the applied heuristic procedure. It determines how close each point in one class is located relative to the points in an adjacent area.



(a)



(b)



(c)

Figure 7. Binary class silhouette for segmentation methods: a – full dataset, b – winter segment after segmentation by the membership function, c – «conditional» winter segment after segmentation by the SEED method

In any given experiment, the dataset was split several times. Silhouette values were grouped by domains. By evaluating the quality of each domain using the silhouette coefficient, it is possible to give a priori estimate of the classifying model [49, 50]. The silhouette coefficient values show that the data obtained using the considered segmentation methods have approximately the same "compactness" properties. Simultaneously, on average, the silhouette coefficient values are better for the segmented samples than for the entire sample, indicating the data uniformity and possible improvement the data processing in segment. The graphs show that in the case of a segmented set, the values are better balanced and form a more compact domain compared to the data of the entire set (Figure 8).

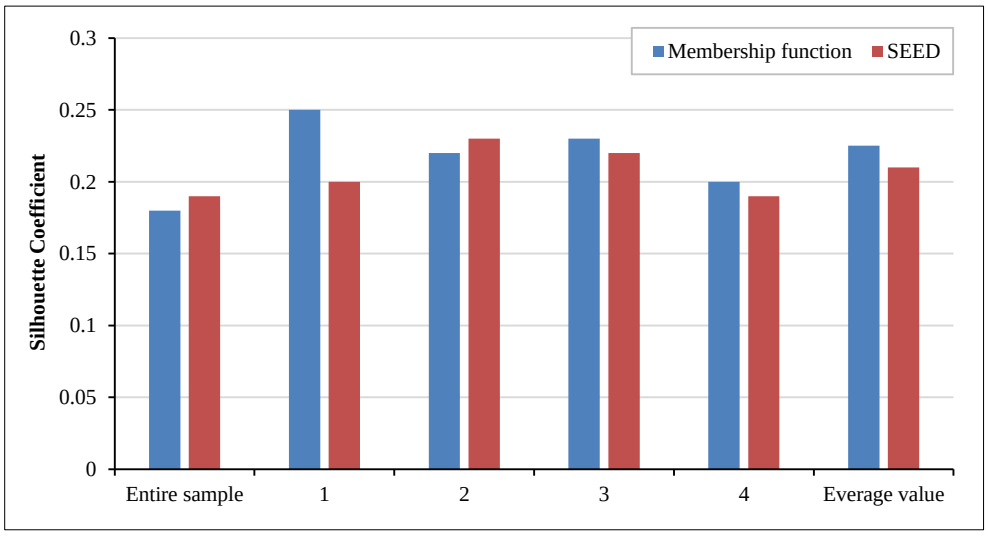


Figure 8. The "Silhouette" values coefficients for four divisions of samples (blue - segmentation based on the membership function, red - segmentation based on the SEED method)

3-3-Algorithm Evaluation

Two divisions were carried out to evaluate changes in performance indicators. The entire sample was split into parts containing energy consumption values based on the indicator function and the concept change detection method. Subsequently, two splitting methods were used to analyze the classifying algorithms. The statistical properties of the predicted target variables change over time. In the cases under consideration, the tuple data values are affected by a predetermined seasonality factor. For this, datasets were specially selected. Various algorithms were chosen to assess the impact of subsets on the quality of the results of machine learning models: the linear discriminant analysis (LD), the quadratic discriminant analysis (QD), the naive Bayes classifier (NB), the k-nearest neighbors (KNN), the decision tree (DT), and the random forest (RF). The influence of sample segmentation on the qualitative indicators of F-measure, accuracy, and AUC for classification and regression tasks was considered. Table 1 gives the results of classifier testing (AUC – area under the ROC curve, accuracy, and F-measure).

Table 1. Results of the classifying algorithms

		Entire sample	Segmentation based on membership functional				Average	Segmentation based on the SEED method				Average
			1	2	3	4		1'	2'	3'	4'	
LD	F measure	0.81	0.84	0.81	0.82	0.81	0.82	0.83	0.82	0.82	0.8	0.82
	Accuracy %	72.30	78.9	73.1	74	72.8	74.70	76	72.5	73.5	72.3	73.58
	AUC	0.74	0.79	0.73	0.75	0.76	0.76	0.73	0.74	0.74	0.76	0.74
QD	F measure	0.81	0.84	0.81	0.82	0.81	0.82	0.83	0.82	0.81	0.81	0.82
	Accuracy %	72.30	78.20	72.2	74.2	73.4	74.50	75.3	72.7	73.3	73	73.58
	AUC	0.74	0.79	0.73	0.75	0.76	0.76	0.74	0.75	0.74	0.77	0.75
KNN	F measure	0.79	0.81	0.51	0.8	0.78	0.73	0.80	0.62	0.79	0.78	0.75
	Accuracy %	70.50	77.60	72.40	72.90	72.30	73.80	75.70	71.00	71.60	72.1	72.60
	AUC	0.71	0.79	0.72	0.74	0.76	0.75	0.76	0.72	0.71	0.76	0.74
NB	F measure	0.79	0.84	0.81	0.81	0.81	0.82	0.83	0.82	0.80	0.80	0.81
	Accuracy %	71.40	76.10	73.3	73.6	72.6	73.90	73.5	73.4	71.5	72.2	72.65
	AUC	0.74	0.78	0.73	0.75	0.76	0.76	0.73	0.74	0.74	0.76	0.74
DT	F measure	0.82	0.84	0.87	0.86	0.85	0.85	0.83	0.87	0.87	0.89	0.87
	Accuracy %	72.46	77.52	77.61	77.63	77.55	77.58	76.96	76.38	76.11	76.55	76.5
	AUC	0.74	0.74	0.76	0.75	0.75	0.75	0.77	0.80	0.78	0.79	0.79
RF	F measure	0.83	0.87	0.88	0.88	0.88	0.88	0.88	0.87	0.88	0.86	0.87
	Accuracy %	72.56	75.58	77.11	74.92	74.54	75.53	74.6	75.1	74.1	74.1	74.5
	AUC	0.80	0.85	0.86	0.85	0.88	0.86	0.84	0.86	0.84	0.85	0.84

There was an increase in classification performance indicators in the analyzed subsamples compared with the classification of the entire training set. The test results show that splitting the total sample into separate subsets improves many classification performance indicators for the selected algorithms. The results in Tables 1 and 2 show that the best results for different methods are achieved in different segments. This allows for selecting the model that has the best processing performance for each segment.

Table 2. The values of performance indicators in segmentation according to the SEED method

		Entire sample	1	2	3	4	5	6	7	8	9	10	11	12	13	Average
LD	F measure	0.81	0.80	0.85	0.81	0.79	0.81	0.85	0.98	0.78	0.81	0.84	0.82	0.80	0.83	0.83
	Accuracy %	72.30	71.30	80.20	72.20	70.40	74.10	80.00	72.20	69.30	73.30	79.20	74.30	71.40	76.90	74.22
	AUC	0.74	0.76	0.82	0.70	0.74	0.77	0.82	0.68	0.74	0.76	0.83	0.76	0.75	0.82	0.77
QD	F measure	0.81	0.79	0.85	0.81	0.79	0.82	0.86	0.81	0.78	0.81	0.83	0.82	0.80	0.83	0.82
	Accuracy %	72.30	72.00	80.30	72.50	71.10	74.10	80.70	72.40	70.00	73.00	78.00	74.50	71.80	77.60	74.46
	AUC	0.74	0.76	0.82	0.70	0.74	0.79	0.82	0.69	0.74	0.76	0.84	0.76	0.76	0.82	0.77
KNN	F measure	0.79	0.79	0.79	0.87	0.84	0.79	0.82	0.87	0.81	0.77	0.81	0.88	0.82	0.80	0.85
	Accuracy %	70.50	70.50	70.80	81.50	75.6	70.2	74.1	81.5	71.8	68.8	72.4	83.3	74.6	70.9	78.90
	AUC	0.71	0.71	0.75	0.82	0.72	0.73	0.78	0.83	0.69	0.74	0.76	0.85	0.74	0.75	0.84
NB	F measure	0.79	0.78	0.81	0.80	0.77	0.79	0.81	0.78	0.76	0.79	0.79	0.80	0.77	0.78	0.79
	Accuracy %	71.40	72.80	75.70	71.40	70.3	73.6	75.9	69.5	70.4	73.5	73.8	73.5	70.8	72.4	72.58
	AUC	0.74	0.75	0.82	0.70	0.74	0.77	0.82	0.68	0.74	0.76	0.83	0.76	0.75	0.82	0.76
DT	F measure	0.82	0.84	0.85	0.86	0.9	0.88	0.88	0.84	0.9	0.83	0.89	0.87	0.91	0.85	0.87
	Accuracy %	72.46	75.50	77.80	81.50	75.6	73.2	74.1	81.5	71.67	78.8	72.4	83.78	77.6	74.9	76.79
	AUC	0.74	0.78	0.75	0.82	0.81	0.73	0.78	0.83	0.79	0.84	0.86	0.85	0.75	0.74	0.79
RF	F measure	0.83	0.87	0.87	0.91	0.93	0.89	0.92	0.89	0.90	0.91	0.94	0.92	0.91	0.90	0.90
	Accuracy %	72.56	77.80	78.70	86.40	79.30	79.60	75.90	79.50	79.40	83.50	83.80	83.50	80.80	82.40	80.81
	AUC	0.80	0.85	0.82	0.80	0.88	0.87	0.82	0.88	0.79	0.89	0.83	0.86	0.85	0.82	0.85

However, there are situations when the proposed method may fail. This loss causes a large scatter of data, complicates the processes of domain splitting, and necessitates the formation of complex separating surfaces for certain types of classifiers.

To solve classification problems in datasets [45–48], additional markup was carried out. Table 3 shows the qualitative indicators for the classification of working and non-working days by power consumption. The average daily output was calculated in the datasets, and two conditional states of "insufficient" and "excessive" electricity generation under various weather conditions were considered relative to this threshold. The results in Table 3 show an improvement in quality scores for segments compared with the whole sample.

Table 3. Qualitative indicators for the classification of working and non-working days by power consumption

		Steel Industry Energy Consumption Dataset				Valencia (Sun energy generation)				Valencia (Wind energy generation)			
		Entire sample	4	6	12	Entire sample	4	6	12	Entire sample	4	6	12
LD	F measure	0.8	0.84	0.86	0.88	0.80	0.82	0.84	0.85	0.81	0.86	0.88	0.89
	Accuracy %	70.3	73.14	74.53	75.8	72.10	75.02	76.45	77.76	71.30	74.19	75.59	76.87
	AUC	0.74	0.75	0.78	0.79	0.71	0.72	0.73	0.73	0.74	0.76	0.76	0.79
QD	F measure	0.81	0.86	0.87	0.88	0.81	0.85	0.85	0.86	0.82	0.86	0.87	0.89
	Accuracy %	71.3	74.2	75.62	76.91	72.30	75.21	76.65	77.97	72.30	75.21	76.64	77.94
	AUC	0.74	0.77	0.78	0.79	0.74	0.77	0.79	0.81	0.72	0.75	0.76	0.78
KNN	F measure	0.7	0.75	0.76	0.78	0.75	0.78	0.79	0.82	0.80	0.84	0.84	0.87
	Accuracy %	70.5	73.35	74.74	76.02	70.50	73.34	74.73	76.01	70.50	73.35	74.75	76.02
	AUC	0.71	0.74	0.75	0.77	0.71	0.72	0.74	0.76	0.71	0.73	0.74	0.75
NB	F measure	0.79	0.81	0.83	0.85	0.79	0.82	0.83	0.85	0.78	0.82	0.84	0.87
	Accuracy %	69.4	72.22	73.59	74.84	70.30	73.14	74.52	75.78	70.40	73.24	74.63	75.89
	AUC	0.72	0.75	0.76	0.79	0.72	0.75	0.76	0.78	0.73	0.76	0.79	0.79
DT	F measure	0.81	0.84	0.86	0.89	0.81	0.84	0.84	0.87	0.82	0.86	0.88	0.91
	Accuracy %	71.4	74.29	75.71	77	71.30	74.18	75.59	76.89	71.90	74.81	76.22	77.5
	AUC	0.75	0.78	0.8	0.82	0.73	0.75	0.75	0.76	0.71	0.73	0.74	0.75
RF	F measure	0.83	0.89	0.89	0.91	0.83	0.87	0.89	0.9	0.82	0.86	0.88	0.91
	Accuracy %	71.5	74.4	75.83	77.12	71.30	74.18	75.59	76.87	71.40	74.28	75.7	76.99
	AUC	0.73	0.74	0.76	0.77	0.74	0.77	0.79	0.79	0.76	0.78	0.8	0.81

It is possible to split the selected dataset further into subsets by increasing the threshold, identifying the concept change for heuristic methods, and using additional information. The proposed solution can be used as an addition to various classification models. A more complex segmentation regarding additional parameters is also acceptable. Such segmentation will improve performance indicators by reducing the phenomenon of data "outliers."

The proposed data preparation technique can be applied to improve regression performance indicators. Various algorithms were chosen to assess the impact of subsets on the quality of the results of machine learning models: linear regression (LR), Gaussian regression (GR), decision trees (DT), support vector machines (SVM), adaptive neurofuzzy inference systems (ANFIS), generalized regression neural networks (GRNN), neural networks with radially basic elements (RBF), and neural networks (ANN).

For each model, the entire sample in full and data from individual segments were used.

The RMSE loss function. A classical single-output regression metric that calculates the absolute difference between the predicted and actual outputs was chosen as a measure for evaluating the regression algorithm:

$$L_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \quad (16)$$

where $y_i = a(x_i)$ is result of the prediction of the selected algorithm; $\hat{y}_i$– actual value of the target variable.

The problem of predicting the electricity generation by power plant. Loss function values for different segments are given in Table 4.

Table 4. RMSE loss function results for different classifiers in 4 segments for membership function and SEED method

RMSE	Entire sample	Segmentation based on membership functional				Average	Segmentation based on the SEED method				Average
		1	2	3	4		1	2	3	4	
LR	6.36	6.55	5.65	5.33	4.60	5.53	7.25	4.93	5.08	4.52	5.45
GR	6.03	6.34	5.60	5.31	4.59	5.46	7.03	4.90	5.06	4.49	5.37
DT	6.37	7.10	6.10	5.77	5.01	5.99	7.83	5.40	5.55	4.97	5.94
SVM	5.90	6.56	5.65	5.33	4.61	5.54	7.26	4.93	5.08	4.53	5.45
ANFIS	3.11	2.95	2.96	2.95	2.94	2.95	2.99	2.90	2.91	2.92	2.93
RBE	2.62	2.47	2.46	2.49	2.49	2.48	2.47	2.47	2.48	2.47	2.47
GRNN	2.59	2.45	2.46	2.45	2.49	2.46	2.45	2.50	2.41	2.49	2.46
ANN	3.10	2.94	2.94	2.97	2.95	2.95	2.92	2.96	2.90	2.90	2.92

In the case of segmentation, on average, there is a decrease in the loss function values compared to the full non-segmented sample. Data segmentation makes it possible to reduce the loss function for different sample areas, to allocate separate segments with a smaller data span, which determines lower values of the loss function on average in the regression problem. The results of the RMSE loss function for different classifiers in different segments number for the SEED method are presented in Table 5.

Table 5. RMSE loss function results for different classifiers in different number segments for the SEED method

	Combined Cycle Power Plant Data Set				Valencia (Sun energy generation)				Valencia (Wind energy generation)			
	Entire sample	4	6	12	Entire sample	4	6	12	Entire sample	4	6	12
LR	4.56	4.43	4.39	4.36	9.42	9.12	9.08	9.02	14.11	13.64	13.56	13.46
GR	4.91	4.76	4.73	4.72	9.12	8.84	8.79	8.75	15.01	14.52	14.44	14.35
DT	4.07	3.95	3.93	3.9	9.15	8.85	8.8	8.75	14.27	13.79	13.73	13.63
SVM	3.94	3.84	3.82	3.8	8.72	8.43	8.41	8.36	14.15	13.68	13.59	13.5
ANFIS	1.99	1.94	1.94	3.83	6.24	6.05	6.03	5.98	10.44	10.1	11.97	9.98
RBE	1.04	1.04	1.03	1.03	5.21	5.05	5.03	5.02	9.34	9.03	8.98	8.95
GRNN	1.27	1.25	1.24	1.25	5.42	5.25	5.23	5.22	9.21	8.9	8.85	8.81
ANN	1.81	1.78	1.79	1.77	7.31	7.08	7.06	7.02	12.61	12.18	12.13	12.05

The allocation of sequence segments of the information flow and the evaluation of their properties allow finding and assigning machine learning methods with the best characteristics. On individual segments, the methods show lower values of the loss function than when processing the entire sample. The results show that the proposed method application, where each data sample segment is assigned a method that has the best quality indicators on it, allows reducing the values of the RMSE loss function from 1 to 8% compared to processing the entire sample.

In the future, to improve quality, it will be possible to use a combination of methods, where each method is assigned to its own segment.

3-4- Results Analysis

A situation occurs when the best achievable quality indicators in each segment and sample show different models. It becomes possible to improve the quality indicators of processing by selecting the algorithm with the best value for each segment. Thus, selecting data segments and evaluating their properties allows the search and assignment of machine learning models with the best characteristics. Similarly, it is possible to compare ensembles consisting of several complex models or elementary algorithms.

In practice, it is not always possible to create various independent models. In the example above, the algorithms are trained on the same sets, reducing their diversity. It is not always possible to realize the division of the training data sample so that the data are random, homogeneous, and independent. As a result, there may be a situation where there is, for example, one "good" and one "bad" algorithm by quality indicators, and this will lead to the ensemble results having a worse quality than those of the "good" algorithm.

Simultaneously, the computational costs of aggregating and training a group of complex predictive models are higher than training a single classifier. This can increase time and computational costs when there is a concept change or a change in the data properties, compared to "substituting" a ready-made model. It is not always possible to build models using different combinations of features, for example, when analyzing one-dimensional rows. And this, in turn, entails the impossibility of achieving their diversity. The average of the models will be an improvement only if the models are independent of each other.

The transformation of data properties can occur in information flows with constant incoming sequence data. As a result, strong classification models trained on historical data may become weak at different time intervals, and vice versa. Such changes in the properties of predictive models occur in a very short period of time, leading to a worse quality of problem-solving by an ensemble of models than that of one classifier.

4- Discussion

4-1- Main Findings of This Study

One of the main problematic issues with machine learning methods is the data processing during the transformation of their properties. Improving the "quality" of processing is achieved by forming complex and relatively resource-intensive models. They are highly labor-intensive and require computational resources for automation. The proposed method is aimed at the segmentation of the data sample and is based on considering the factors that influence the changes in the ranges of target variables. Automatic implementation of segmentation is possible with the help of models and methods for detecting points of concept change and drift. The identification of effects makes it possible to form segmented data samples based on current situations. It is possible to select and assign a pre-trained model for each resulting segment, depending on the data properties.

4-2- Comparison with Other Studies

One way to improve quality is to use models based on refined local information [24]. The analysis is conducted on individual predictors that have the maximum impact on processing quality. However, quality data tuples can be obtained under the influence of various factors [27]. After some time, the transformation of their properties is possible, which will require additional analysis. Pre-training the models on the segments and evaluating the properties of the obtained sample segments makes it possible to assign the most efficient algorithms and classification models for each subset. Assigning a particular algorithm with the best qualitative indicators to a segment allows us to obtain an increase in various quality indicators for each classifier from 1% to 8%. This is comparable to the results of the quality indicators of ensemble models [11, 12]. However, unlike the proposed method, they require complex aggregation functions and computing resources for the parallel operation of the data processing models.

In the proposed solution, it is possible to select a separate, best-quality pre-trained model for each segment to avoid the cost of aggregating the results of ensemble methods [7, 11]. The changes in data properties provide an opportunity to assign a model choice quickly. The proposed method can be applied as an addition to complex data processing models to perform the segmentation of sequences first to improve the qualitative performance of its constituent algorithms.

4-3- Implications and Explanation of the Findings

Data-sample segmentation provides an opportunity to reduce the loss function of individual segments. The search algorithm for trend change points allows the selection of individual segments with a smaller data range, which determines lower values of the loss function on average. Highlighting segments of data stream sequences and evaluating their properties allows us to identify machine learning models with the best performance. On individual segments, algorithms show lower loss function values than when processing the entire sample. By considering the loss function, it is possible to assign to a segment of the model the best value. Pre-training samples with similar properties can reduce the time for model preparation. An analysis of the model results and the actual values of the sequence can be applied to generate training data to refine the model. Hierarchies are further possible when the top-level model is applied to assign the most efficient lower-level model to an individual segment. The proposed solution aims at further improving and extending ensemble methods and hybrid classifiers. It represents a functional engineering technique that improves the quality of individual elements of a data processing model by partitioning the set into subsets.

5- Conclusion

To improve the quality of the models' performance, it is possible to implement pre-processing for data sampling. In different analyzed segments, it is necessary to implement separating surfaces of various complexity, leading to better performance of different models on different subsamples. Collecting observation objects is time-consuming, and various shifts in the values of individual parameters can occur within the tuples. Feature extraction may lose its relevance if concept drift occurs. In this regard, it is necessary to process incoming data samples and analyze each segment continuously. Information about data properties in the segments strongly depends on how the sample is segmented and separated. The processing of this data is necessary to obtain information about class separability, to form a separating surface, and to improve the quality performance of the classifying algorithm.

Using several models to improve the quality of prediction results in the form of ensemble methods leads to the fact that despite the various combinations combining individual algorithms into a model, situations occur where such a combination may not only not improve but even worsen the result. It is necessary to prevent such situations, which the proposed solution facilitates.

6- Declarations

6-1- Author Contributions

Conceptualization, I.L. and M.S.; methodology, I.L.; software, I.L.; validation, I.L., and M.S.; investigation, I.L. and M.S.; resources, M.S.; data curation, I.L.; writing—original draft preparation, M.S.; writing—review and editing, I.L.; visualization, M.S.; supervision, I.L.; project administration, I.L.; funding acquisition, M.S. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available in the article.

6-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2019). Learning under Concept Drift: A Review. IEEE Transactions on Knowledge and Data Engineering, 31(12), 2346–2363. doi:10.1109/TKDE.2018.2876857.

[2] Marinin, M., Karasev, M., Pospehov, G., Pomortseva, A., Kondakova, V., & Sushkova, V. (2023). Comprehensive study of filtration properties of pelletized sandy clay ores and filtration modes in the heap leaching stack. Journal of Mining Institute, 259, 30–40. doi:10.31897/pmi.2023.7.

[3] Blyth, C. R. (1972). On Simpson's Paradox and the Sure-Thing Principle. Journal of the American Statistical Association, 67(338), 364. doi:10.2307/2284382.

[4] Tsai, S.-Y., & Chang, J.-Y. (2018). Parametric study and design of deep learning on leveling system for smart manufacturing. 2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE). doi:10.1109/smile.2018.8353980.

[5] Dang, Q., & Yuan, J. (2023). A Kalman filter-based prediction strategy for multiobjective multitasking optimization. Expert Systems with Applications, 213(B), 119025. doi:10.1016/j.eswa.2022.119025.

[6] Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M. L., Chen, S. C., & Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys, 51(5), 1–36. doi:10.1145/3234150.

[7] Pedersen, T. (2000). A simple approach to building ensembles of naive Bayesian classifiers for word sense disambiguation. arXiv Preprint cs/0005006. doi:10.48550/arXiv.cs/0005006

[8] Sethi, T. S., & Kantardzic, M. (2018). Handling adversarial concept drift in streaming data. Expert Systems with Applications, 97, 18–40. doi:10.1016/j.eswa.2017.12.022.

[9] Cerqueira, V., Torgo, L., & Soares, C. (2019). Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters. ArXiv. doi:10.48550/arXiv.1909.13316.

[10] Liu, C., Fu, L., Li, H., & Chen, B. (2023). Dynamic Prediction Algorithm for Low-Voltage Distribution Network Power Loss in a Smart City Based on Classification Decision Tree and Marketing Data. Journal of Testing and Evaluation, 51(3), 20220096. doi:10.1520/JTE20220096.

[11] Ye, X., & Zhao, J. (2023). Heterogeneous clustering via adversarial deep Bayesian generative model. Frontiers of Computer Science, 17(3), 173322. doi:10.1007/s11704-022-1376-2.

[12] Zhou, Z. H., & Feng, J. (2019). Deep forest. National Science Review, 6(1), 74–86. doi:10.1093/nsr/nwy108.

[13] Muksin, U., Riana, E., Rudyanto, A., Bauer, K., Simanjuntak, A. V. H., & Weber, M. (2023). Neural network-based classification of rock properties and seismic vulnerability. Global Journal of Environmental Science and Management, 9(1), 15–30. doi:10.22034/gjesm.2023.01.02.

[14] Kim, J. Y., Kim, D., Li, Z. J., Dariva, C., Cao, Y., & Ellis, N. (2023). Predicting and optimizing syngas production from fluidized bed biomass gasifiers: A machine learning approach. Energy, 263, 125900. doi:10.1016/j.energy.2022.125900.

[15] Park, J., & Kim, S. (2020). Machine Learning-Based Activity Pattern Classification Using Personal PM2.5 Exposure Information. International Journal of Environmental Research and Public Health, 17(18), 6573. doi:10.3390/ijerph17186573.

[16] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. doi:10.1109/TPAMI.2016.2577031.

[17] Oikarinen, E., Tiittanen, H., Henelius, A., & Puolamäki, K. (2021). Detecting virtual concept drift of regressors without ground truth values. Data Mining and Knowledge Discovery, 35(3), 726–747. doi:10.1007/s10618-021-00739-7.

[18] Takacs, A., Toledano-Ayala, M., Dominguez-Gonzalez, A., Pastrana-Palma, A., Velazquez, D. T., Ramos, J. M., & Rivas-Araiza, E. A. (2020). Descriptor Generation and Optimization for a Specific Outdoor Environment. IEEE Access, 8, 52550–52565. doi:10.1109/ACCESS.2020.2975474.

[19] Widmer, G., Kubat, M. (1993). Effective learning in dynamic environments by explicit context tracking. In: Brazdil, P.B. (eds.) Machine Learning: ECML-93. ECML 1993. Lecture Notes in Computer Science, vol 667. Springer, Berlin, Heidelberg. doi:10.1007/3-540-56602-3_139.

[20] Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi:10.1145/502512.502529.

[21] Black, M., & Hickey, R. J. (1999). Maintaining the performance of a learned classifier under concept drift. Intelligent Data Analysis, 3(6), 453–474. doi:10.3233/IDA-1999-3604.

[22] Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., ... & Song, D. (2019). Efficient task-specific data valuation for nearest neighbor algorithms. arXiv Preprint arXiv:1908.08619. doi:10.48550/arXiv.1908.08619.

[23] Klinkenberg, R. (2001). Using labeled and unlabeled data to learn drifting concepts. Workshop notes of the IJCAI-01 Workshop on Learning from Temporal and Spatial Data, 16-24. Held in conjunction with the International Joint Conference on Artificial Intelligence (IJCAI): AAAI Press, 4-6 August, 2001, Menlo Park, United States.

[24] Liu, Y., Liu, Y., Yu, B. X. B., Zhong, S., & Hu, Z. (2023). Noise-robust oversampling for imbalanced data classification. Pattern Recognition, 133. doi:10.1016/j.patcog.2022.109008.

[25] Maletzke, A., Dos Reis, D., Cherman, E., & Batista, G. (2019). DyS: A Framework for Mixture Models in Quantification. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 4552–4560. doi:10.1609/aaai.v33i01.33014552.

[26] Li, P., Wu, X., & Hu, X. (2012). Mining recurring concept drifts with limited labeled streaming data. ACM Transactions on Intelligent Systems and Technology, 3(2). doi:10.1145/2089094.2089105.

[27] Djouzi, K., Beghdad-Bey, K., & Amamra, A. (2022). A new adaptive sampling algorithm for big data classification. Journal of Computational Science, 61, 101653. doi:10.1016/j.jocs.2022.101653.

[28] Nan, L. (2013). Classification algorithm for data streams with concept drift and its applications. Master Thesis, Fujian Normal University, Minhou, China.

[29] Wang, G., & Wang, Y. (2023). Self-attention network for few-shot learning based on nearest-neighbor algorithm. Machine Vision and Applications, 34(2), 28. doi:10.1007/s00138-023-01375-5.

[30] Wu, Z., Efros, A. A., & Yu, S. X. (2018). Improving generalization via scalable neighborhood component analysis. In Proceedings of the european conference on computer vision (ECCV), 685-701. doi:10.48550/arXiv.1808.04699.

[31] Maillo, J., Ramírez, S., Triguero, I., & Herrera, F. (2017). kNN-IS: An Iterative Spark-based design of the k-Nearest Neighbors classifier for big data. Knowledge-Based Systems, 117, 3–15. doi:10.1016/j.knosys.2016.06.012.

[32] Deng, Z., Zhu, X., Cheng, D., Zong, M., & Zhang, S. (2016). Efficient KNN classification algorithm for big data. Neurocomputing, 195, 143–148. doi:10.1016/j.neucom.2015.08.112.

[33] Ou, G., He, Y., Fournier-Viger, P., & Huang, J. Z. (2022). A Novel Mixed-Attribute Fusion-Based Naive Bayesian Classifier. Applied Sciences (Switzerland), 12(20), 10443. doi:10.3390/app122010443.

[34] Karegowda, A. G., V, P., Jayaram, M. A., & Manjunath, A. S. (2012). Rule based Classification for Diabetic Patients using Cascaded K-Means and Decision Tree C4.5. International Journal of Computer Applications, 45(12), 45–50. doi:10.5120/6836-9460.

[35] Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. Mechanical Systems and Signal Processing, 107, 241–265. doi:10.1016/j.ymssp.2017.11.024.

[36] Maletzke, A. G., dos Reis, D. M., & Batista, G. E. A. P. A. (2018). Combining instance selection and self-training to improve data stream quantification. Journal of the Brazilian Computer Society, 24(1). doi:10.1186/s13173-018-0076-0.

[37] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), 1–37. doi:10.1145/2523813.

[38] Huang, D. T. J., Koh, Y. S., Dobbie, G., & Pears, R. (2014). Detecting Volatility Shift in Data Streams. 2014 IEEE International Conference on Data Mining. doi:10.1109/icdm.2014.50.

[39] Zheng, X., Aragam, B., Ravikumar, P. K., & Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning. 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

[40] Di Franco, G., & Santurro, M. (2021). Machine learning, artificial neural networks and social research. Quality and Quantity, 55(3), 1007–1025. doi:10.1007/s11135-020-01037-y.

[41] Scanagatta, M., Corani, G., Zaffalon, M., Yoo, J., & Kang, U. (2018). Efficient learning of bounded-treewidth Bayesian networks from complete and incomplete data sets. International Journal of Approximate Reasoning, 95, 152–166. doi:10.1016/j.ijar.2018.02.004.

[42] Trevizan, B., Chamby-Diaz, J., Bazzan, A. L. C., & Recamonde-Mendoza, M. (2020). A comparative evaluation of aggregation methods for machine learning over vertically partitioned data. Expert Systems with Applications, 152, 113406. doi:10.1016/j.eswa.2020.113406.

[43] Guo, Y., Chen, Q., Chen J., Wu Q., Shi Q., Tan M. (2019). Auto-embedding generative adversarial networks for high resolution image synthesis. IEEE Transactions on Multimedia, 21(11), 2726-2737. doi:10.1109/TMM.2019.2908352.

[44] Lee, M. H., Kim, N., Yoo, J., Kim, H. K., Son, Y. D., Kim, Y. B., Oh, S. M., Kim, S., Lee, H., Jeon, J. E., & Lee, Y. J. (2021). Multitask fMRI and machine learning approach improve prediction of differential brain activity pattern in patients with insomnia disorder. Scientific Reports, 11(1), 9402. doi:10.1038/s41598-021-88845-w.

[45] Zhu, X. (2010). Power Supply dataset. Stream Data Mining Repository: Florida Atlantic University. Available online: http://www.cse.fau.edu/~xqzhu/stream.html (accessed on April 2023).

[46] Kaggle (2023). Energy generation dataset. Available online: https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather/data?select=energy_dataset.csv (accessed on April 2023).

[47] UCI. (2023). Steel Industry Energy Consumption Dataset. Machine Learning Repository: Center for Machine Learning and Intelligent Systems. Available online: http://archive.ics.uci.edu/ml/datasets/Steel+Industry+Energy+Consumption+Dataset (accessed on April 2023).

[48] UCI (2014). Combined Cycle Power Plant Data Set. Machine Learning Repository: Center for Machine Learning and Intelligent Systems https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant# (accessed on April 2023).

[49] Meiseles, A., & Rokach, L. (2020). Source Model Selection for Deep Learning in the Time Series Domain. IEEE Access, 8, 6190–6200. doi:10.1109/ACCESS.2019.2963742.

[50] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53–65. doi:10.1016/0377-0427(87)90125-7.