Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 9, No. 5, October, 2025

Adaptive Segmentation of Information Sequences for Machine Learning Modular Regression Models

Ilya Lebedev [1], Mikhail Sukhoparov [2*], Viktor Semenov [1], Dmitry Khasanov [1]

[1] *St. Petersburg Federal Research Centre of the Russian Academy of Sciences, St. Petersburg, 199178, Russian Federation.*

[2] *Russian State Hydrometeorological University (RSHU), St. Petersburg, 192007, Russian Federation.*

Abstract

The research objective is to construct an adaptive model for modular machine learning structures that improves the processing quality of information sequences. The novelty of the proposed methodology is that it can identify segments of an information sequence obtained using various methods and assign models with the best quality indicator values to subsequences. Classical methods allow tuning of the model to the entire data sample. The improvement consists of the proposed solutions that consider the inverse problem of forming segments of data sequences, such that their properties correspond to the processing model. The proposed methodology was tested on various models and datasets. Segmentation and assignment of regression models with the best characteristics to individual segments allow the reduction of the mean square error (MSE) and mean absolute error (MAE) to 8%. The findings show an opportunity to increase of 5-8% for weak LR, SVM, and GR models, while strong DT, CNN, ANN, ANFIS, and XGBoost models improve by 1-4% in scenarios with limited data. Segmentation enables more efficient training and reduces sensitivity to noise and outliers. The proposed solution allows the selection of segmentation strategies and model combinations considering local data properties. Its application enables the implementation of existing machine learning architectures to improve the quality of training and analysis of information sequences and increase adaptability, scalability, and interpretability.

Keywords:

Machine Learning;
Adaptive Models;
Processing Quality Improvement;
Regression Models;
Mean Square Error (MSE);
Mean Absolute Error (MAE).

Article History:

Received:	30	May	2025
Revised:	15	August	2025
Accepted:	02	September	2025
Published:	01	October	2025

1- Introduction

Analyzing information sequences obtained during the operation of heterogeneous observation objects is a complex process [1]. Time series and information sequences play important roles in various fields. The analysis of financial markets, the state of information systems, climate change, energy generation, and various types of activities necessitates the development of effective models and methods for processing time series that involve solving classification and forecasting problems. Improving the quality indicators of data processing is one of the fundamental tasks of machine learning (ML) methods. The main direction of the solution is associated with the formation of effective data processing models. The achievable values of the quality indicators of basic processing algorithms, such as the naive Bayes classifier, linear discriminant, and decision trees, depend on the properties of the processed samples. Simple models may lose their adequacy in the event of changes in distributions, event frequency, and trends. The data sequence can be divided into separate segments to improve productivity and processing quality [2]. Segmentation can provide a good approximation for data processing models. However, several issues related to the choice of segments must be solved for efficient implementation. Sequence length, number of segments, outliers, noise, and data properties can significantly affect the results. There are several possible partitioning options [3].

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

DOI: http://dx.doi.org/10.28991/ESJ-2025-09-05-08

When processing regression dependencies, situations often appear when the relationship between the response and explanatory variables changes sharply at particular points [4]. For example, in sociology, these are different slices of public opinion; in medicine, the risks of diseases may differ significantly on the age of people; in power engineering, the generation of electricity may depend on climate and temperature conditions; a sharp increase in information and service messages may occur at specific external impacts in the analysis of the state of information systems and networks. Methods of detecting discord are used to identify the points of sharp change in the relationship between variables. In the first case, the task is to find the most appropriate model for forecasting or searching for extreme values [5–7]. In the second case, patterns between the response and explanatory variables in intervals limited by change points must be identified.

Sequence segmentation is one of the most significant tasks for improving the quality of regression analysis. Segments whose information properties do not correspond to data processing models can significantly deteriorate processing results. The solution to this problem depends on the processing of noise, outliers, and abnormal values of the data sample, which cannot always be unambiguously interpreted. Segmentation helps adapt to data and reduces computational complexity as data volume increases.

This study aims to improve the quality of information sequence processing by using efficient modular, adaptive, and universal computational structures with data sample partitioning to form subsequences with optimal properties and information structures for training and applying regression models. The novelty is in the methodology for selecting segments of an information sequence obtained using different methods by distinguishing the quality functional of regression processing models. The proposed solution's peculiarity is the segment selection for the ML algorithm, which in practice allows for more efficient adjustment of the model to the data.

2- Literature Review

Regression and forecasting problems have been solved using various ML techniques. The most popular processing model is the neural network (NN). Convolutional neural network (CNN), Recurrent neural network (RNN), GAN, and DNN architectures have been used to solve tasks, and these architectures are constantly being improved [8]. New neural network models, such as the KAN, have the properties of adaptability and flexibility of tuning to solve subject-oriented regression problems, have great opportunities for expansion and scalability, and ensure high data processing speed. However, their use causes some problematic issues of computational complexity, memory limitations when training on large datasets related to "vanishing" and "explosive" gradients, training stability, and the need to build data processing and storage architecture when solving large-scale data processing problems. Despite the advantages of the capacity to compute hidden patterns, a large amount of data is still required to train a neural network, which is not always available.

In general, a neural network is a superposition of functions, where neurons and their sets play the role of functions [9-11]. Within this framework, an alternative is to replace the superposition of functions based on neurons with functions. These are data processing models that implement naive Bayes algorithms, decision trees, support vectors, or other neural network structures. The sequence of their use leads to the same problem solution as that with a complex construction neural network. It is possible to obtain a general model with better interpretability of results, less resource-intensive, and achieving results commensurate with the neural network by defining the rules of application of less complex and less resource-intensive models separately and selecting the most appropriate algorithms depending on the data properties and current processing processes. However, selecting and building such models is a nontrivial task that requires optimizing the processes that occur.

The models can optimize and select the data sequence properties. The optimization process has two main directions. They aim to improve the quality of processed data and build an efficient processing model [12].

Feature space formation methods can improve the data processing quality. Among them are approaches that perform data separation based on clustering, search for time-series disagreement points, and detect "concept drift." The problem of finding these points with the changing properties of predictors and target variables is essential for solving many applied problems using stepwise (SSR). Many methods, such as Bayesian analysis, maximum likelihood method, quantile regression, and nonparametric and analytical methods, have been applied for this purpose. However, the complexity of an approximating function passing through such points often leads to high computational and resource costs when searching for this function.

Many researchers have considered sampling separation processes as an auxiliary in machine learning methods. These depend on the type and properties of the processed information. Several basic directions exist for the processing of information sequences. For example, Rinaldo et al. [13] used classical clustering methods. Despite their relative simplicity, they improve the processing quality of data with specific properties. In high-dimensional environments, when more than one change point exists, detection and localization problems arise. Data segmentation methods that employ dynamic programming [14-16], binary segmentation, and Bayesian methods [17] can be used to solve this problem.

Bayesian approaches for regressions on change points were proposed in previous studies [17-19], but the solutions described in them have high computational complexity and require numerous iterations when using the Monte Carlo method to model Markov chains. Golzari Oskouei et al. [20] used statistical thresholds for sequence segmentation. The outliers and noise of the observed objects must be considered in this solution to improve processing accuracy. Bardwell & Fearnhead [21] presented a segmented regression method using linear transition to estimate the regression model's change points. Wang et al. [22] proposed deep learning methods. The qualitative performance of data processing depends on resource intensity and data analysis. Huang et al. [23] used an extended Dickey-Fuller test for data stationarity to identify segments. Lu & Chang [24] reviewed a pattern-based method to identify time-series segments. Tallman and West [25] proposed an approach based on maximum likelihood search by applying a dynamic programming method. Studies [26, 27] have implemented a method of simultaneous multiscale change point estimation (SMUCE), which simultaneously determines the points and confidence intervals. Wang et al. [28] proposed a binary-segmentation method for sample partitioning.

Table 1 presents the characteristics of the methods considered, prerequisites for their application, and emerging difficulties.

Table 1. Characteristics of the sequence segmentation methods considered

Segmentation methods	Characteristics	
	Advantages	**Disadvantages**
Classical clustering methods	Relatively easy to implement	Strong dependence on the data properties
Dynamic programming methods	Useful when the target function is a non-differentiable, discrete variable change	More resource-intensive than other methods
Binary segmentation methods	Simple implementation and high speed of processing	A high probability of erroneous segmentation
Bayesian methods	Implementation on small sample sizes and high-dimensional data	Preliminary description subjectivity, which can lead to weak qualitative indicators
Method based on statistical sequence segmentation thresholds	High sensitivity and proven mathematical apparatus	The quality of the results depends on data analysis and interpretation (determining and interpreting outliers and noises of observation objects is necessary).
The segmented regression method	Simplicity of the computational algorithms	The accuracy of change point detection decreases as the data width increases; this method is sensitive to data variability (variance shift).
Deep learning methods	Achieved high-quality indicators with an adequate model	Dependent on the initial properties of the sequence; complicated model building; resource-intensive; and requires numerous iterations
Segmentation method using the extended Dickey-Fuller test (ADF)	Adapted for series with stationarity	The complexity increases significantly in the case of the non-stationary series. There is also a need to check several points of discrepancy
Template method for identifying time-series segments	Multiple similarity and dissimilarity metrics were used.	Dependent on the quality of templates and selected similarity and dissimilarity criteria
Simultaneous multiscale change point estimation (SMUCE)	The number of jumps, their location, and confidence intervals for disagreement points are estimated.	Requires solving a resource-intensive optimization problem in which a sufficiently large number of local solutions may exist at intervals.

Segmentation is one of the areas for improving the quality of data processing. This article discusses the selection of segmentation methods and assignment of models to segments using a pre-selected quality indicator. Its application allows for assigning models with the best performance in individual segments in ensemble ML methods. As a result, it is possible to compensate for the individual shortcomings of individual models while minimizing the impact of random errors. The selection of segmentation methods, assignment of models to a segment, and assessment of the achieved quality indicator make it possible to consider the local properties of the data. This can be important under resource constraints because training an individual algorithm on a local segment reduces computational costs compared to training a complex model on the entire sample. In the case of data property transformation, training on a local segment allows for faster model reconfiguration. Data segmentation using control of quality indicators of processing models allows for assigning different areas of competence to different algorithms, selecting a model that has the best performance on the data of a separate segment, and determining the internal structure of the data for further analysis and processing. For the effective use of segmentation in ensemble methods, the properties of the data must be evaluated under various external and internal influences.

Data segmentation in ML impacts the performance of models. It allows models to focus on specific data subsets, which often leads to more accurate predictions or classifications. However, the impact of segmentation depends on the type of model and the tasks it is used for. The internal structure of the data is determined for further analysis, processing, investigation of probable relationships between observation objects, and discovery of new patterns. They aim to reduce variance and improve generalizability [29]. Data partitioning, considering the imbalance, and the distribution of observation objects help improve the SVM, RF, and XGBoost models' quality indicators [30].

The second direction is related to the search for the most efficient DPM. Basic algorithms, such as linear regression, logistic regression, the machine vector method, and neural network approaches, are used in simple problems. The obtained values of the processing quality indicators depend on the processed samples' properties. The presence of outliers, data linearity, and variables' independence significantly affected the quality indicators of the processing results.

Approaches aimed at forming ensembles of models and algorithms that combine several machine-learning methods have been used to overcome these disadvantages and improve the quality of information sequence processing [31, 32]. Table 2 summarizes the main ensemble methods and their characteristics defined in Mohammed & Kora [32].

Table 2. Characteristics of the main ensemble methods

Main Ensemble Methods	Advantages	Disadvantages
Bagging	Reduces the impact of single model errors, thereby improving overall accuracy. Reduces sensitivity when transforming data properties and allows parallel training of processing models	In the case of "wrongly" selected models, bagging can worsen the result. Significant computational resources are required when training complex models
Boosting	Effective for dealing with unbalanced data samples	Prone to overtraining with weak models; demonstrates significant computational complexity and results in complex compositions that are difficult to tune when solving practical problems.
AdaBoost	Robust to noisy and unbalanced data	Sensitive to outliers
Gradient Boosting	More robust to errors and missing data than most other models	Extremely sensitive to outliers and wastes enormous resources in their presence
XGBoost	Best performance with regular and dense data	Irregular, sparse data without observations and significant gaps between observations
Stacking	Enables the creation of multilevel models	Prone to overtraining; demonstrates significant increase in computational complexity as the number of models increases; requires model usage
Hybrid Ensemble	Relatively high accuracy and versatility when processing samples with different properties	Sensitive to data quality and quantity; it is complicated to tune models

In addition to the aforementioned ensemble methods, complex multilevel models can be built using sampling systems, weighted voting, rule-based decision-making systems, a priori knowledge of the data, cascades of simple models, and deep neural networks [33]. One of the directions for the formation of effective computational structures can be used, for example, to level the problem of choosing the distance between switching regions, performing statistical tests to determine the most appropriate trend, and knowing the change points in piecewise regression. The main disadvantages of the algorithm structures are the complexity of training, aggregation of results, resource intensity, and increased running time. Improperly selected models and methods of aggregating their results can deteriorate the overall forecast.

Lebedev & Sukhoparov [34] used a multi-model approach that forms an ensemble of models and algorithms that combines several machine-learning methods to overcome the identified problematic issues of improving the quality of regression sequence processing. Rule-based decision-making systems, a priori knowledge of the data, weighted, sampling-based voting systems that evaluate models based only on their past prediction performance [35], cascades of simple algorithms, and deep neural networks [36] can be applied.

Transformer and AutoML technologies are actively developed based on similar solutions, and models of the modular approach and mixture of experts (MoE) are improved. However, along with significant advantages in data processing, they have some limitations, being "black boxes," and require significant computational resources to function and store knowledge and databases. Despite some success, the problem of sparse data remains for these technologies. Simple models, such as depth-constrained decision trees, can better address this problem.

All modern methods focus on processing information with specific properties. Their main disadvantages are the computational complexity of training, resource intensity, and the need to use external databases and knowledge, which are significant limitations in implementing local, embedded, and autonomous systems. In the case of the transformation of data properties without the organization of constant learning processes, the model may lose adequacy over time [37, 38].

A methodology that uses data-sampling partitioning to form subsequences with optimal properties and information structures to train and apply regression models must be developed. This methodology aims to improve the processing quality, which is ensured by the computational structures formed with its help, combining modularity, allowing the replacement of regression models, the possibility of transformation, providing a change in the computational structure, leveling the errors of some models at the expense of others, and having the ability to interpret the results and form the composition based on the requirements of computing resources.

3- Methodology

3-1- Characteristics of the Proposed Method

One of the main challenges in solving sequence-processing problems is building an optimal model that makes it possible to achieve high-quality processing performance. This study involved the development of a methodology for the adaptive segmentation of information sequences and the assignment of regression models to segments. The proposed methodology and other methods based on shortest separation sequence (SSS) can deal with outliers and noise and reduce the time required for model preparation and training. Figure 1 illustrates the proposed solution.

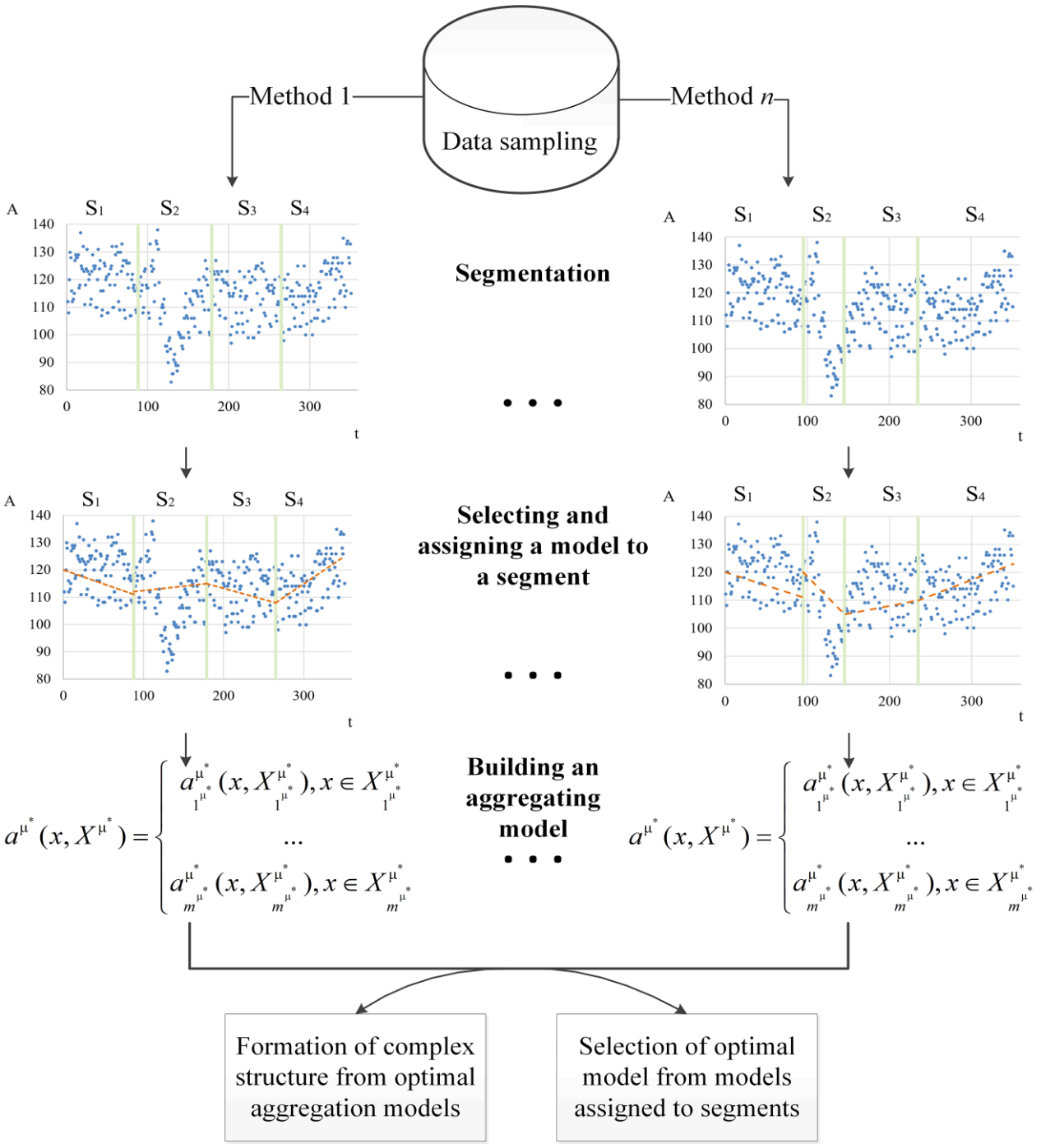

Figure 1. Conceptual diagram of the proposed methodology's process sequence

Initially, the processing models and segmentation methods were defined. Data sequences were segmented using different methods. Models were trained on the segments. Each segment was assigned the processing model with the best value for the selected quality indicator. Consequently, an aggregation model can be implemented for each segmentation method that contains the segments' best-quality indicator models. Furthermore, depending on the problem to be solved and the constraints, we select a segmentation method and an optimal aggregation model or build a parallel, cascade, hierarchical structure of selected methods and models, similar to ensemble methods.

Classical ML approaches adjust the learning models to the properties of the training sample. The proposed solution considers the inverse problem of forming segments of data sequences such that their properties correspond to the processing model.

This study proposes a methodology for forming information subsequences based on the quality indicators of the processing models. Its peculiarity is tuning machine learning models to the processed data by dividing the sequence into segments and selecting the segmentation method such that the properties of the data obtained in the segment best fit the processing model. In contrast to the classical approach, the analysis and selection of segments of the data sequence are

formed for the model when the model is tuned to the data in the proposed methodology. When applied to regression and forecasting problems, building a range of aggregation models for different methods of segmenting a single sequence makes it possible to build multilevel models and ensembles that reduce processing errors and sensitivity to outliers.

The methodology focuses on building modular structures where some functions can be replaced with others. Any segmentation method can be used to divide a sequence. Simple algorithms and more complex NN structures, ensembles, and hybrid models can be used. It provides an opportunity to scale depending on available resources and required quality indicators to determine the composition of processing models, pre-select segments describing the subject area for customizing the model and its use in local low-power devices or high-performance systems, and mitigate certain disadvantages of various current technologies. These disadvantages include the transformer's resource requirements, AutoML tuning complexities, and sensitivity to sparse XGBoost data. Depending on the goals and requirements for solving regression problems, a situation may arise when operating without the integration of simpler models is sufficient.

Different sequence segmentation methods result in segments with information that possesses different properties. The segment properties depend on the number of observation objects and their distributions, trends, and periodicity. Various models can be used to process segments. The quality indicators of the model depend on the segment information properties. Some models show the best results on segments for one method of sequence partitioning, whereas other models when choosing another method of partitioning or changing its parameters.

In the proposed solution, the segmentation method is linked to a model that defines a processing model with a sequence segmentation method. The proposed methodology aims to process segments with any properties, and the quality is ensured by assigning a model. The proposed solution does not gravitate toward dividing the data using any specific method. If one model was unsuitable for segment processing, another model with a higher quality index value was applied. If the aggregated model, composed of the models assigned to the segments, has insufficient quality indicators, another known segmentation method can be used, and the achieved quality indicator values may be analyzed.

However, the proposed solution works under conditions where the properties of the general population on the segment remain unchanged. If this condition is met, the processed sequence size decreases. Segments can be selected to reduce outliers and data noise. Simpler models can be used to process them. They are easier to select based on the segments' properties. However, if the properties are distorted when forming segments or, for example, if the number of observation objects in the segment is too small, the model will lose its adequacy, and the results will deteriorate.

The first step is to select a method to segment the sequence while using a predetermined quality index of the processing model. The next step is to build an aggregate model consisting of the algorithms that achieve the segment's best quality indexes.

3-2- Problem Setting

To implement the methodology, we define a sample of observation objects X, processing models $\{a_1,\dots,a_N\} \in A$, and data segmentation methods $\{\mu_1,\dots,\mu_L\} \in \Omega$. Information sequence X is divided into separate segments. This results in a set of partitioning methods X^{μ} where the segments are processed by models a_i. We need to find method μ^* to divide the sequence into segments $X^{\mu^*} = \{X_{1^{\mu^*}}^{\mu^*},\dots,X_{m^{\mu^*}}^{\mu^*}\}$ and assign a processing model $a_i \in A$ with the best value of the quality functional $Q(a_i(x), X_{i^{\mu^*}}^{\mu^*}) \rightarrow \max\limits_{a_i \in A, \mu^* \in \Omega}$ to each segment.

The mean square error (MSE) and mean absolute error (MAE) metrics can be used as quality functions for regression estimation depending on the problems to be solved. Thus, the main focus is to assign the best quality models for segment processing, which in some cases makes it possible to pay less attention to implementing computationally complex regression data segmentation procedures that consider the properties of the observation objects in the data sequences.

On the one hand, the precondition of the proposed methodology is the assumption of data heterogeneity, and on the other hand, the arising ambiguity, for example, in high-dimensional conditions where more than one point may exist in the localization region. The observation objects of a regression sequence can form different regions where trends and data properties change, resulting in segments where the trend direction may change or where a linear relationship may exist in some cases and a non-linear relationship in others. Consequently, different functions better approximate different subsequences within X.

Considering the partitioning methods $\mu_l \in \Omega$ of the data sequences $\{X_{1_{\mu_l}}^{\mu_l},\dots,X_{m_{\mu_l}}^{\mu_l}\} \in X^{\mu_l}$ and obtaining different combinations of segments, it is necessary to choose the partitioning method μ^* and determine the aggregable processing function a^* consisting of the models whose use achieves the best quality indicators for the segments:

$$a^*, \mu^* = \underset{\mu_l \in \Omega, a_i \in A}{\operatorname{argmax}} Q(a_i(x, X^{\mu_l})) \tag{1}$$

By determining the sample partitioning method and assigning the algorithm with the highest performance according to the training results to each segment compared to the other algorithms, we obtain an aggregated sequence processing model. The best-predefined segmentation methods and number of segments were selected to build the ML models.

3-3- Implementation of Methodology

The methodology is implemented in several stages. The first step is to select the basic processing algorithms. Attention is focused on implementing learning processes to ensure the diversity of models solved using various data manipulations, forming feature spaces, samples, and fundamentally different processing models. The next step is to analyze the underlying models, which usually requires the exclusion of models that do not achieve high prediction accuracy. Subsequently, we processed the obtained results.

Figure 2 shows the proposed methodology's general algorithmic sequence.

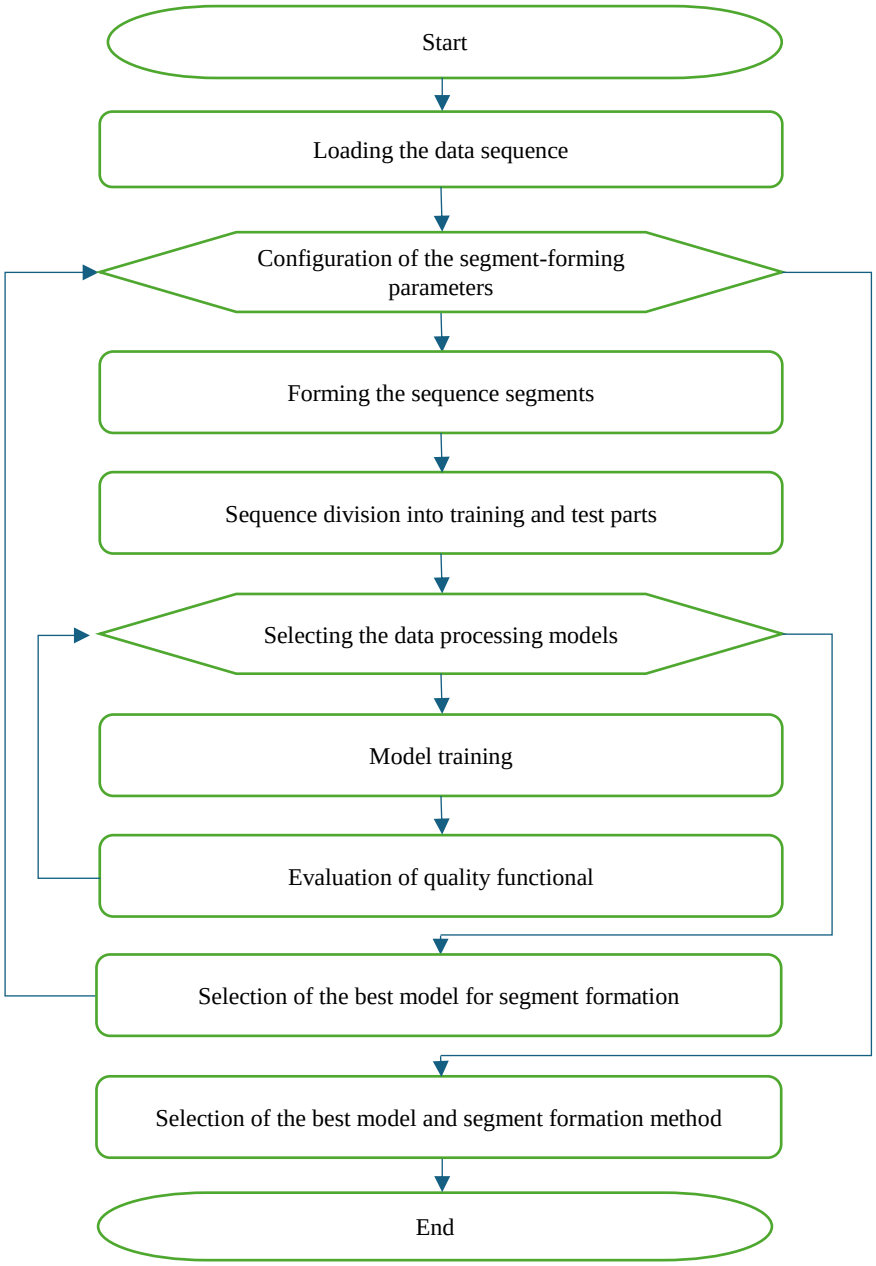

Figure 2. Flowchart of the general algorithmic sequence of actions

The proposed methodology implementation involves ten steps to configure the models and perform their predefined tasks.

Step 1: Search for methods of sample partitioning $\{\mu_1, \ldots, \mu_L\} \in \Omega$.

Step 2: The number of segments is increased to a certain maximum value.

Step 3: Processing sample X by partitioning method μ_l, forming segments $\{X_{1_{\mu_l}}^{\mu_l}, \ldots, X_{j_{\mu_l}}^{\mu_l}, \ldots, X_{m_{\mu_l}}^{\mu_l}\} \in X^{\mu_l}$ for this method and for the number of segments m, j – segment ordinal number.

Step 4: Search for the number of segments.

Step 5: Search for processing models.

Step 6. Training models a_i on segments $X_{j_{\mu_l}}^{\mu_l}$.

Step 7: If not all models have been trained, return to step 5; otherwise, determine the best of the models $\{a_1,\dots,a_N\} \in A$ on the segment $X_{j_{\mu_l}}^{\mu_l}$ by the value of the model quality factor $a^{j_{\mu_l}} = \underset{a_i \in A}{\operatorname{argmax}} Q(a_i(x, X_{j_{\mu_l}}^{\mu_l}))$.

Step 8: If not all segments have been processed, return to step 4; otherwise, determine the sample $X_m^{\mu_l} = \{X_{1_{\mu_l}}^{\mu_l},\dots,X_{m_{\mu_l}}^{\mu_l}\}$ for the selected method and the selected number of segments and form an aggregation model from the models $\{a^{1_{\mu_l}},\dots,a^{m_{\mu_l}}\} \in A$, showing the best results in quality factor value for m segments on the sample $X_m^{\mu_l}$ after processing by the method μ_l $a_m^{\mu_l}(x, X_m^{\mu_l}) = \begin{cases} a^{1_{\mu_l}}\left(x, X_{1_{\mu_l}}^{\mu_l}\right), x \in X_{1_{\mu_l}}^{\mu_l} \\ \dots \\ a^{m_{\mu_l}}\left(x, X_{m_{\mu_l}}^{\mu_l}\right), x \in X_{m_{\mu_l}}^{\mu_l} \end{cases}$.

Step 9: If the maximum number of segments has not been reached, return to step 2. Otherwise, determine the number of segments of the μ_l method sample X partitioning at which the best quality factor has been achieved, and determine the sample segments and the model that achieves the best quality factor $m^{\mu_l} = \underset{m \in \{1,\dots,M\}}{\operatorname{argmax}} Q(a_m^{\mu_l}(x, X_m^{\mu_l}))$ and determine the sample segments $X_{m^{\mu_l}}^{\mu_l} = \{X_{1^{\mu_l}}^{\mu_l},\dots,X_{m^{\mu_l}}^{\mu_l}\}$ and the model that achieves the best quality factor when partitioned by the method μ_l $a_{m^{\mu_l}}^{\mu_l}(x, X_{m^{\mu_l}}^{\mu_l}) = \underset{a_m^{\mu_l} \in A}{\operatorname{argmax}} Q(a_m^{\mu_l}(x, X_m^{\mu_l}))$.

Step 10. If there are still sequence partitioning methods left, then return to step 1. Otherwise, the final selection of the sample partitioning method is made, where the maximum quality factor $\mu^* = \underset{\mu_l \in \mu}{\operatorname{argmax}} Q(a_{m^{\mu_l}}^{\mu_l}(x, X_{m^{\mu_l}}^{\mu_l}))$ is achieved, and the number of segments obtained by the selected sample partitioning method $m^{\mu^*} = \underset{m \in \{1,\dots,M\}}{\operatorname{argmax}} Q(a_{m^{\mu_l}}^{\mu_l}(x, X_{m^{\mu_l}}^{\mu_l}))$, determine the segments of the sample processed by the selected partitioning method $X^{\mu^*} = \{X_{1^{\mu^*}}^{\mu^*},\dots,X_{m^{\mu^*}}^{\mu^*}\}$, and finally form the aggregation processing model $a^{\mu^*}(x, X^{\mu^*}) = \begin{cases} a_{1^{\mu^*}}^{\mu^*}\left(x, X_{1^{\mu^*}}^{\mu^*}\right), x \in X_{1^{\mu^*}}^{\mu^*} \\ \dots \\ a_{m^{\mu^*}}^{\mu^*}(x, X_{m^{\mu^*}}^{\mu^*}), x \in X_{m^{\mu^*}}^{\mu^*} \end{cases}$.

Implementing the methodology involves performing actions to form an information sequence and defining the processing methods. Each partitioning method has certain parameters, such as the number of partitions. For simplicity, we can assume that methods with different parameters are partitioning methods. All methods receive the input data sample X and form segments $X_{1_{\mu_l}}^{\mu_l},\dots,X_{M_{\mu_l}}^{\mu_l}$. Data can be one-dimensional or multidimensional information sequences depending on the problem to be solved. The partitioning method $\mu_l \in \Omega$ forms a set of sample segments $\{X_{1_{\mu_l}}^{\mu_l},\dots,X_{M_{\mu_l}}^{\mu_l}\} \in X^{\mu_l}$.

All predetermined sequence processing models were trained on the observation objects of each set of segments $X_{1_{\mu_l}}^{\mu_l},\dots,X_{M_{\mu_l}}^{\mu_l}$ formed by the partitioning method $\mu_l \in \Omega$, and all predefined sequence processing models $\{a_1,\dots,a_N\} \in A$ were trained. Subsequently, the system is regulated, where for each partitioning method $\mu_l \in \Omega$ of all trained models $\{a_1,\dots,a_N\} \in A$ on the segments $X_{1_{\mu_l}}^{\mu_l},\dots,X_{M_{\mu_l}}^{\mu_l}$ obtained with its application; the values of the quality functional $Q(a_i(x, X_{j_{\mu_l}}^{\mu_l}))$ are compared. Each partitioning method $\mu_l \in \Omega$ on each segment $X_{j_{\mu_l}}^{\mu_l}$ corresponds to the best quality factors.

Based on expression (1), a partitioning method $\mu_l \in \Omega$ is selected where the best models show the best results on the segments, forming tuples $< a_{j_{\mu_l}}, X_{j_{\mu_l}}^{\mu_l} >$ where a model is assigned to a segment.

The solution exhibits exponential computational complexity. Many repetitive routine operations necessitate the analysis of the calculation results at separate stages of model formation to eliminate unpromising resource-intensive processes. The results of the methodology application are the model formation and the choice of the segmentation method. If the model loses its adequacy, the methodology can be launched independently and in parallel with the system operation.

The methodology is aimed at use in situations where heuristic methods could prove challenging. For example, in the case of multidimensional data, hidden patterns or ambiguous situations can be interpreted in different ways. In the proposed solution, segment selection is performed automatically using the quality functional.

If sufficient resources are available, several sequence segmentations can be considered to form ensembles of aggregation models, thereby reducing sensitivity to errors. The availability of resources under the restriction of the selected models, number of segments, and segmentation methods makes it possible to use the methodology by organizing a parallel estimation process for changing data properties and forming segmentation methods and processing models on its basis. The proposed solutions can be improved using continuous learning methods (CLM) [39-41]. Figure 3 shows the element interrelationship framework for the implementation of self-learning technology. Certain actions for training and model formation can operate in parallel.

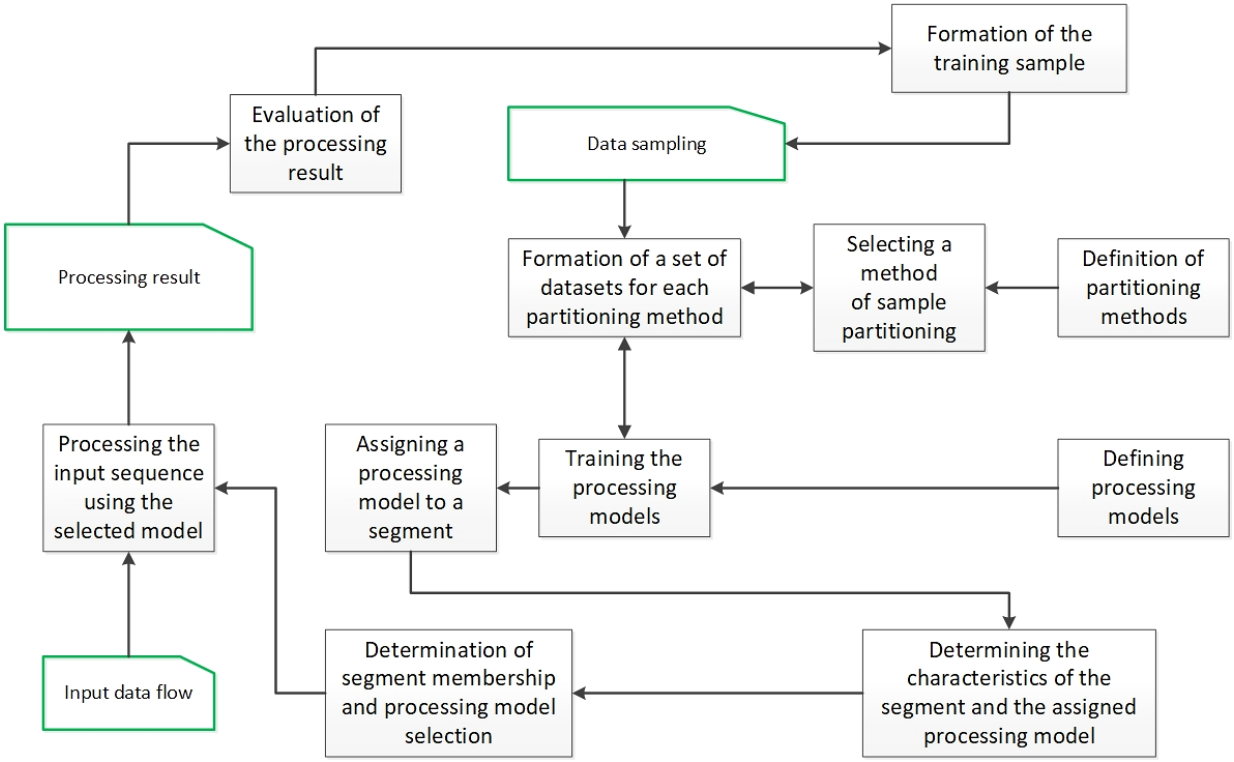

Figure 3. Framework of element interrelationships

The incoming information flow is then processed. The properties of the observed objects were analyzed and calculated. They correlate with segments, and a predetermined model is selected with the best values of the quality function on a similar segment of the training sample for their processing. The model's predicted results were compared with the actual values of the observed objects obtained from the recording systems and devices. If the errors increase above a predetermined threshold, a decision is made to form a data sample.

4- Results

4-1- Experimental Setup

The experiment aimed to evaluate the change in qualitative indicators of different models of information sequence processing when segmentation methods are applied to the considered methodology. The realized experimental evaluation of the models and the data are available in Semenov [42] using two low-complexity segmentation approaches. In one case, a supervised approach with segment boundaries based on an optimal change-point detection algorithm with linear computational cost was used [43]. In the second case, we used an approach that divides the sequence into segments equal to the number of observations. Segmentation using k-means was considered for the multidimensional dataset data.

The proposed methodology was evaluated for predicting one-dimensional and multidimensional data sequences, and the qualitative indicators were compared across different datasets. The input data format represents an information sequence for one-dimensional data [44], and the dimensionality of the dataset data determines this format for multidimensional data. In the first case, the information sequence of the observation objects contained *m* equal parts, and in the second case, *m* segments using information about the change points. The output data format represents the predicted regression values. Simple cross-validation covered each segment without normalization and outlier processing. The sample included two parts: training and testing. The number of runs was 10. The parameters of the models are determined by the implementation and standard environment settings [42, 45]. Table 3 summarizes the study's main characteristics.

Table 3. Characteristics of the experimental models

Models	Characteristics
LR	Free and linear terms for each predictor.
SVM	The linear kernel function.
GR	The quadratic exponential kernel function
DT	Binary tree. The tree depth depends on the training sample.
CNN	Activation function (reLU), hidden layer size is 10.
ANN	Activation function reLU, the hidden layer size is 8.
ANFIS	Two membership functions for each variable, three selections in the fuzzy logic domain, 10 training iterations.
XGBoost	The maximum tree depth is 5, the number of trees is 100, the learning rate is 0.1,

The optimization of model parameters did not change and was constant during data processing because only the segmentation effect on the loss function results was evaluated. The experiment used datasets with processability by relatively simple models; therefore, the high adaptability architectures and models, such as Transformer and KAN, were not considered in the experiment.

The algorithmic sequence of actions during the experiment is shown in Figure 4.

Input data:

X – Initial test sample;
μ_l – Sample partitioning methods;
a_i – Data processing models LR, SVM, GR, DT, CNN, ANN, ANFIS;
$L_{mse}(a_i, X_{j_{\mu_l}}^{\mu_l})$, $L_{mae}(a_i, X_{j_{\mu_l}}^{\mu_l})$ – MSE and MAE loss function;
M – Maximum number of segments;
L – Number of methods

Output data:

q_msei – MSE function value
q_maei – MAE function value

begin

For l = 1, …, L // Search for partitioning methods
  Form = 1, …, M // Search for the number of segments
  μ_l: $X \to \{X_{1_{\mu_l}}^{\mu_l}, \ldots, X_{j_{\mu_l}}^{\mu_l}, \ldots, X_{m_{\mu_l}}^{\mu_l}\} \in X^{\mu_l}$ // Sample processing
    For j = 1, …, m // Search for the number of segments
      For i = 1, …, N // Search for models
        $q\_mse_i = L_{msei}(a_i, X_{j_{\mu_l}}^{\mu_l})$,
        $q\_mae_i = L_{maei}(a_i, X_{j_{\mu_l}}^{\mu_l})$, // Loss function calculation

end

Figure 4. Pseudocode of the experiment

The values of $\hat{y}_i$ obtained using the segment-assigned model were calculated and compared with the true value of y_i. Quality indicators for data processing were determined using MSE and MAE metrics.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (2)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (3)$$

where $\hat{y}_i$ is the predicted value, y_i is the true value, and n is the number of objects observed.

MSE highlights large errors and allows the selection of a model that produces fewer gross forecast errors. However, if one or more outliers produce a significant error, squaring will lead to the false conclusion that the entire model is performing poorly. MAE is less sensitive to outliers than MSE. This makes it a preferred option when outliers are present in the data but should not significantly affect the model's overall performance. MAE and MSE are not universal metrics by themselves; they should be considered based on the tasks being solved and the data's properties. The proposed methodology gives preference to models and a segmentation method that achieve the best values of a pre-selected quality indicator.

4-2- Experiment on one-Dimensional Data

The experimental data for the one-dimensional regression were obtained from the dataset [44]. Sequence values ranged from 0 to 4500. The sequence was segmented using two methods. Figure 5 shows an example of segmentation into four segments equal to the number of observation objects (Figure 5-A) and an example of segmentation by the algorithm for optimal change-point detection (CPD) with linear computational costs with four segments S_1-S_4 (Figure 5-B).

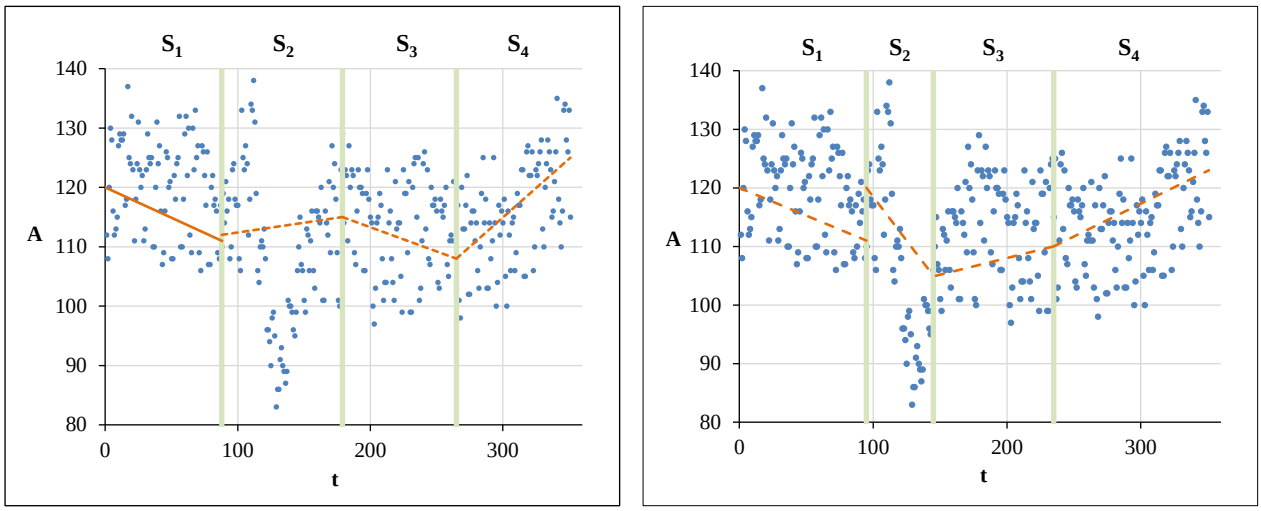

Figure 5. Segmentation of the sequence on the axes "time t - value A" into four segments S_1, S_2, S_3, and S_4 ((a) equal parts; (b) distribution by algorithm for optimal change point detection with linear computational costs)

Figure 5 shows that different information sequences appear in the segments because of different methods of segmentation. The spread of values, distributions, and number of observation objects may differ significantly. These variations could have a significant impact on the training processes and quality indicators achieved by the different models.

Figure 6 summarizes the Equation MSE 2 and Equation MAE 3 values in each segment and whole sample processing for each model.

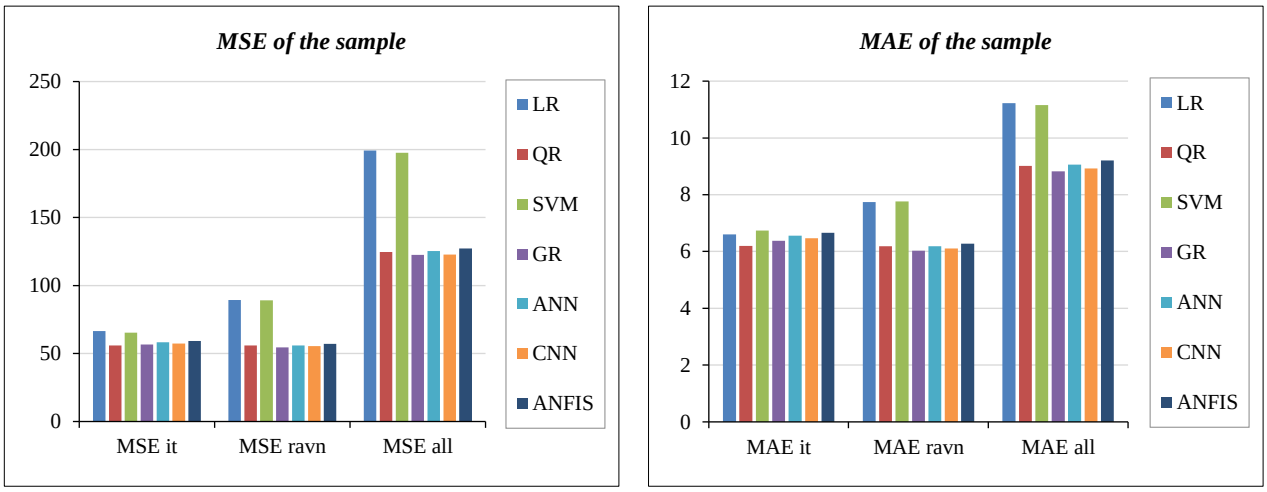

Figure 6. MSE and MAE loss function when approximating the whole sample and when divided into four segments by the equal number of observation objects (ravn) and the algorithm for optimal change point detection with linear computational costs (it).

The obtained MSE and MAE values demonstrate the positive impact of segmentation using both the first and second methods. The loss function values were lower for all considered models after segmentation than for the entire sample. For the case under consideration, segmentation using the optimal change point detection algorithm with linear computational costs was preferable to segmentation into an equal number of observation objects. The compact location and sensitivity to trend changes influence the simplicity of the processing models. Segmentation improves the quality metrics of the models for the given data. They can capture the nuances and patterns specific to each segment, resulting in improved overall performance.

Figure 7 shows the MSE and MAE values of the processing models for each of the four segments. The information properties of the segments change when partitioning the sequence using the selected segmentation methods, resulting in different quality indicators for the processing models. This parameter can be used to assign the processing model to the segment with the best values.

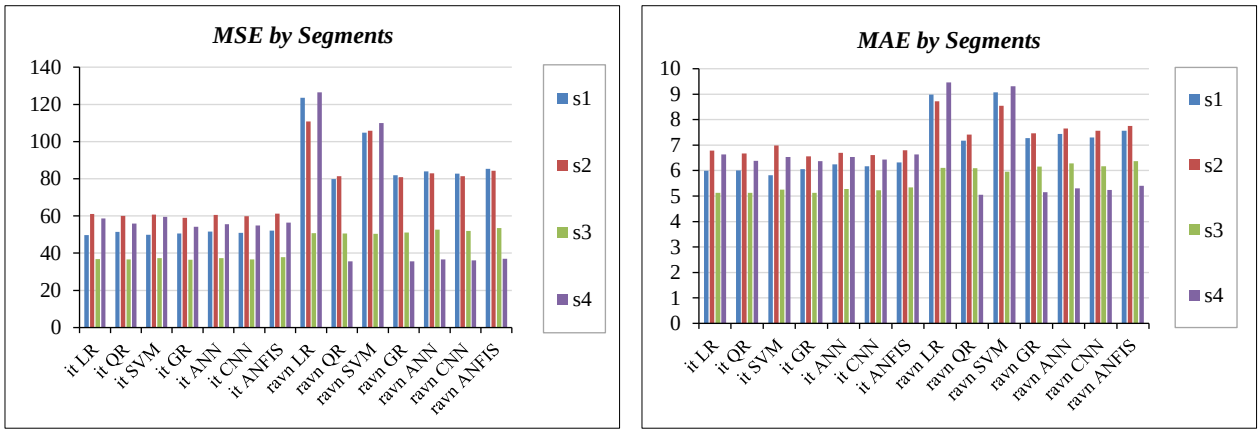

Figure 7. The MSE and MAE values of the algorithms for segments were obtained using an equal number of observation objects (ravn) and the algorithm for optimal change point detection with linear computational costs (it)

In different segments, due to differences in the properties of the objects of observation, the processing models show different values of quality indicators. This is a characteristic of machine learning tasks, in which models are trained on a predetermined set of data. Considering each segment separately, we can identify models with better quality indicators. However, when implementing processing models, it is necessary to track possible changes in data properties that can decrease quality indicators. The quality indicators of the model achieved during the processing depend on the number of segments. Figure 8 shows the dependence of the average MSE and MAE values of the models on the number of segments obtained for the two partitioning methods. When assessing the effect of the number of segments m on the MSE and MAE values, the number of segments increased from 1 (the entire sample) to 35. The values of the MSE and MAE loss functions for the models and partitioning methods were estimated.

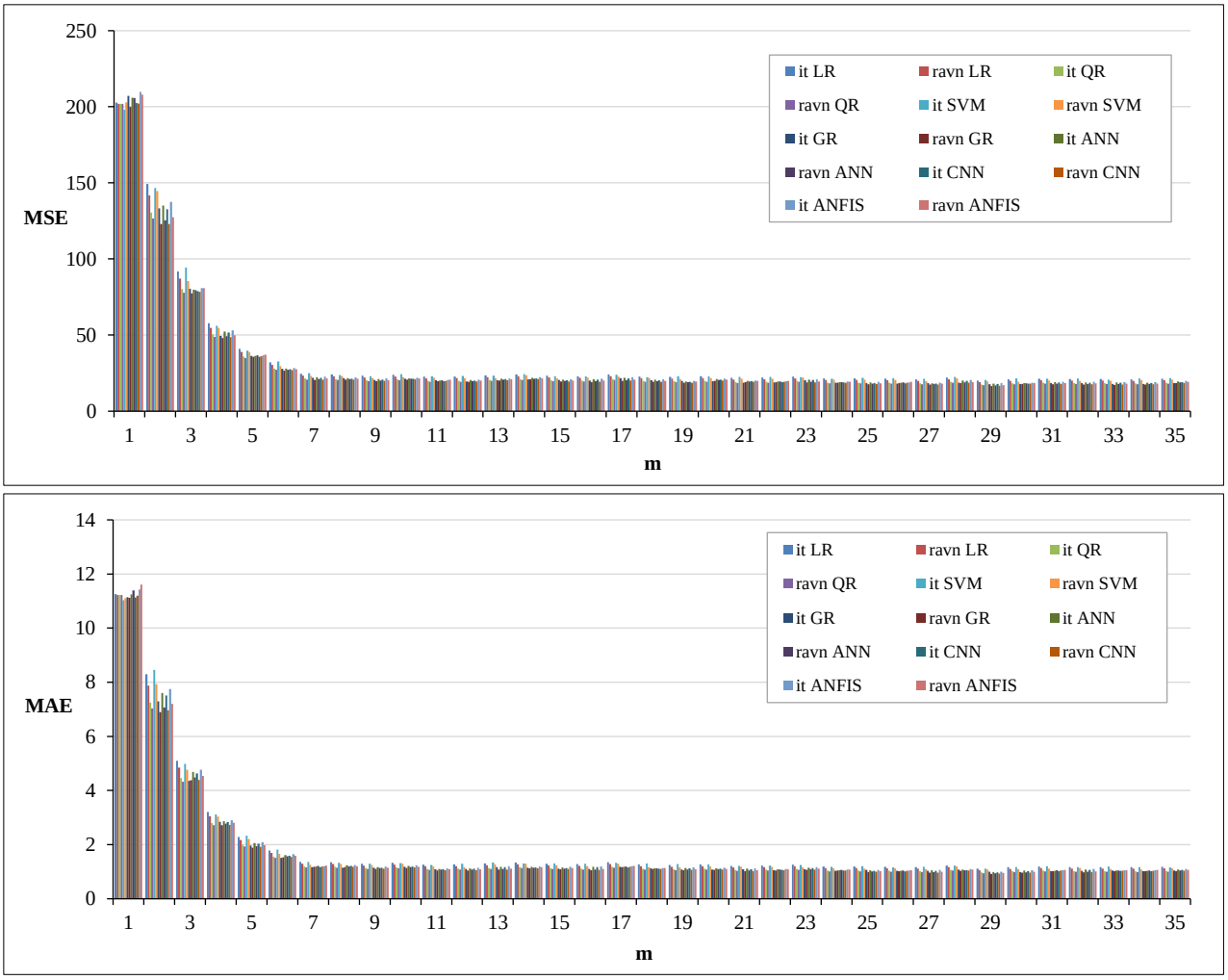

Figure 8. Dependence of MSE and MAE values on the number of segments m for the linear regression obtained by dividing into equal parts according to the number of objects (ravn) and the algorithm for optimal change point detection with linear computational costs.

Figure 8 shows that the number of segments influences the MSE and MAE values. The presented data show the average loss function values per processing model across varying segments. The loss function values decrease as the number of segments increases. The more segments created, the smaller the MSE and MAE become for the selected models and segmentation methods. When the loss function values of the models reach a plateau, an increase in the number of segments does not significantly improve the quality function. If the number of segments grows unlimited, microsegments containing insufficient observation objects may compromise model adequacy. One of the solutions to limit the minimum number of objects may be the application of various heuristic rules; for example, the number of objects of observation should be ten times more than the model parameters. However, this does not guarantee that the objects in this area repeat the properties of the general population, which can lead to ambiguous model estimation results.

4-3- Experiment on Multidimensional Data

In most practical cases, the analyzed data are multidimensional, which imposes specific difficulties in their preliminary processing when detecting outliers and noise and in the unambiguous interpretation of emerging phenomena. Multiple regression was applied to evaluate the quality indicators obtained when segmenting a multidimensional data sequence. The sample from Xhu [44] was used as the study dataset.

Figure 9 shows plots of variations in MSE (m) and MAE (m) depending on the number of segments *m* for the LR, SVM, GR, DT, CNN, ANN, and ANFIS algorithms. The plots show that for most relatively simple models, such as LR, SVM, and GR, segment reduction without considering the properties of the objects contained therein can increase the processing quality factors.

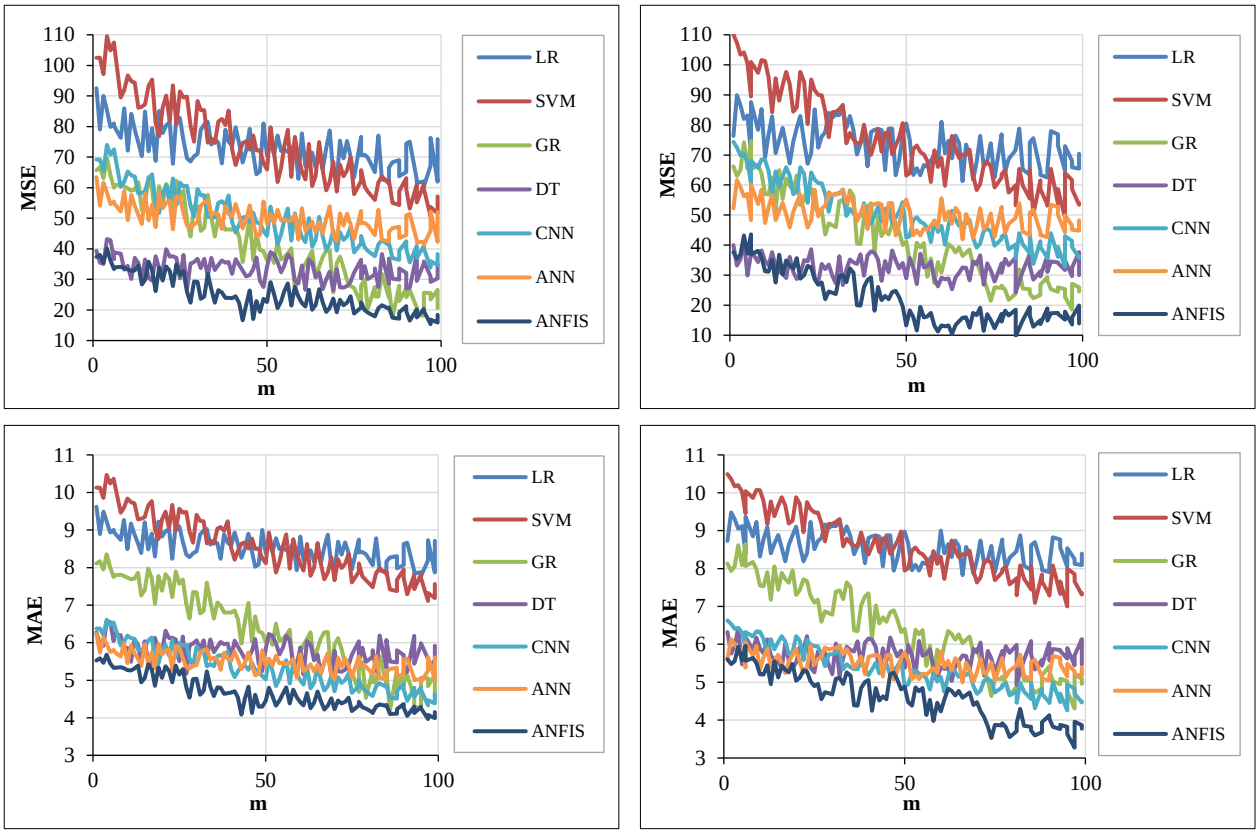

Figure 9. Dependence of MSE and MAE values of different algorithms on the number of segments m for multiple regression when segmenting by division into equal segments (a) and using the algorithm for optimal change point detection with linear computational costs (b).

Data segmentation affects different data processing models differently due to variations in segmentation methods, segment information properties, and algorithm configurations. It did not have a strong effect on individual models. For example, DT exhibits minimal sensitivity because node-level partitioning inherently optimizes locally, whereas segmentation does not guarantee global optimality. Reducing the volume of data and sample imbalance affects neural network models. In the case of an existing linear dependence of the data sequence, it has a small effect on linear regression. However, the proposed segment size reduction methodology did not achieve this effect for all models. For example, for the data sample under consideration, reducing the segment size has a weaker effect on the DT algorithm than on the other algorithms. The MSE and MAE values for this algorithm decreased significantly.

4-4- Experiment on Dataset Values

The experiment involved the analysis of a series of different sets of multiple regression data. Samples [46-48] were used as the experimental data. The effect of segmentation on quality indicators Equations 2 and 3, was evaluated. The sample was partitioned into 15 and 30 segments by the algorithm for optimal change-point detection with linear computational costs by segmenting into an equal number of observation objects, and the MSE and MAE loss functions were determined for each model by the k-means algorithm. Table 4 presents the prediction results of segments and the entire sample.

Table 4. Sample processing (modeling performed by the authors based on [46-48])

Dataset	Segmenting method	Model	MSE			MAE		
			Whole sample	15 segments	30 segments	Whole sample	15 segments	30 segments
Dataset 1	The change point detection algorithm	LR	76.32	74.97	72.4	8.74	8.69	8.67
		SVM	110.11	93.33	90.01	10.49	9.66	8.71
		GR	66.18	55.37	50.45	8.14	7.44	6.87
		DT	39.98	38.46	37.66	6.32	5.52	5.49
		CNN	34.36	32.96	31.7	5.63	5.51	5.42
		ANN	32.26	32.09	31.56	5.62	5.52	5.49
		ANFIS	27.67	27.19	26.81	5.61	5.12	4.93
		XGBoost	26.12	26.1	26.01	5.59	5.11	4.9
	Dividing into equal segments	LR	92.52	84.84	68.54	9.62	9.21	8.28
		SVM	112.51	97.25	89.35	10.12	9.34	8.96
		GR	67.7	56.86	54.09	8.11	7.41	6.64
		DT	39.39	36.48	35.2	6.28	6.04	5.93
		CNN	36.2	34.83	34.14	6.18	5.84	5.59
		ANN	33.43	32.14	31.89	6.13	5.85	5.81
		ANFIS	27.33	26.85	26.41	5.52	5.06	4.93
		XGBoost	26.11	26.09	26.01	5.51	5.05	5.04
	K-means	LR	84.43	81.91	70.47	9.28	9.08	8.83
		SVM	106.31	90.3	85.18	10.39	9.55	8.93
		GR	65.94	55.12	47.28	8.19	7.43	6.83
		DT	39.7	36.5	35.46	6.35	6.18	5.99
		CNN	35.78	34.9	33.42	6.17	6	5.51
		ANN	35.85	35.12	34.23	6.01	5.78	5.47
		ANFIS	33.5	32.04	31.61	5.66	5.2	4.79
		XGBoost	30.62	29.99	27.04	5.43	5.19	5.03
Dataset 2	The change point detection algorithm	LR	0.0327	0.031	0.0282	0.0049	0.0042	0.0039
		SVM	0.0631	0.0618	0.0518	0.0127	0.0089	0.0085
		GR	0.0334	0.0332	0.0331	0.0059	0.0053	0.0049
		DT	0.0258	0.0208	0.0204	0.0039	0.0033	0.0029
		CNN	0.0298	0.0297	0.0296	0.0042	0.0041	0.004
		ANN	0.0381	0.0375	0.0368	0.0051	0.0049	0.0048
		ANFIS	0.0235	0.0232	0.0231	0.0038	0.0035	0.0032
		XGBoost	0.0231	0.023	0.0229	0.0031	0.003	0.0029
	Dividing into equal segments	LR	0.0429	0.0425	0.0423	0.0063	0.0059	0.0055
		SVM	0.0754	0.0669	0.0661	0.0143	0.0118	0.0106
		GR	0.0438	0.042	0.0419	0.0067	0.0064	0.0061
		DT	0.0354	0.0349	0.0343	0.0051	0.0048	0.0047
		CNN	0.0332	0.0329	0.0328	0.0049	0.0047	0.0046
		ANN	0.0311	0.0307	0.0301	0.0047	0.0046	0.0045
		ANFIS	0.0261	0.0246	0.0241	0.0041	0.0039	0.0038
		XGBoost	0.0257	0.0241	0.024	0.0039	0.0036	0.0035
	K-means	LR	0.0462	0.045	0.0436	0.0073	0.0071	0.0068
		SVM	0.0808	0.0778	0.0744	0.0189	0.0159	0.0158
		GR	0.0429	0.0417	0.0409	0.0064	0.0057	0.0055
		DT	0.0359	0.035	0.0349	0.0058	0.0052	0.0051
		CNN	0.0334	0.0331	0.0329	0.0053	0.0051	0.005
		ANN	0.0321	0.0319	0.0318	0.0049	0.0048	0.0047
		ANFIS	0.0311	0.031	0.0309	0.0044	0.0043	0.0042
		XGBoost	0.0279	0.0269	0.0267	0.0032	0.0031	0.003

Dataset 3	The change point detection algorithm	LR	6.84	6.44	6.05	0.7	0.68	0.65
		SVM	9.63	9.06	8.53	0.83	0.81	0.77
		GR	4.66	4.38	4.12	0.56	0.55	0.53
		DT	3.68	3.47	3.36	0.44	0.43	0.41
		CNN	3.58	3.37	3.26	0.43	0.42	0.4
		ANN	3.52	3.4	3.39	0.43	0.42	0.41
		ANFIS	3.39	3.37	3.36	0.39	0.38	0.37
		XGBoost	3.31	3.29	3.28	0.37	0.36	0.35
	Dividing into equal segments	LR	11.56	10.89	10.24	0.91	0.89	0.86
		SVM	16.28	15.32	14.41	1.08	1.05	1.01
		GR	7.87	7.41	6.97	0.73	0.7	0.68
		DT	6.22	5.86	5.52	0.57	0.55	0.54
		CNN	6.04	5.68	5.35	0.56	0.54	0.53
		ANN	5.93	5.58	5.25	0.55	0.53	0.49
		ANFIS	5.87	5.52	5.2	0.55	0.49	0.47
		XGBoost	5.6	5.44	5.41	0.46	0.39	0.38
	K-means	LR	7.08	7.07	6.45	0.57	0.55	0.54
		SVM	12.79	11.77	9.54	0.92	0.89	0.77
		GR	5.06	4.49	3.86	0.49	0.47	0.46
		DT	4.71	4.09	3.09	0.36	0.35	0.34
		CNN	4.33	4.1	3.99	0.34	0.33	0.32
		ANN	3.25	3.19	3.18	0.31	0.29	0.24
		ANFIS	3.19	3.18	3.17	0.29	0.23	0.21
		XGBoost	3.14	3.1	3.07	0.23	0.22	0.2

The data were verified using the Wilcoxon t-test. The null hypothesis H_0 is as follows: "Post-experiment performance is less than pre-experiment performance," and the alternative hypothesis H_1 is as follows: "Post-experiment performance exceeds pre-experiment performance." For the data groups in Table 3, pairwise comparisons of the MSE and MAE column values show $T_{emp} < T_{cr}$ at a significance level of $p \leq 0.01$. Thus, H1 is rejected. The comparison shows that the experimental data segmentation of the sample, in most cases, improves the data-processing quality indicators Equation 2 and 3.

The results in Table 4 show that reducing the sequence segment size mainly reduced the MSE and MAE loss functions by 3-8%. Segmentation changes the data properties. Segmentation affects weak models more significantly than strong models. The results show an opportunity to increase by 5%-8% for weak models, whereas an improvement of 1%–4% is observed for strong models in scenarios with limited data.

Linear models work better in the short term, whereas nonlinear models require large training datasets and resources to achieve good results. If the sequences in the segments replicate the properties of the population and have sufficient training data, then strong models, such as XGBoost, are rather good at identifying patterns in the data. In this case, the quality score may be affected by the reduction of outliers and noise in the observations in the segments during segmentation, which may not have a significant impact on the achieved quality score of a model trained on the entire sample. The manifestation of nonlinearity and the appearance of objects treated as outliers within a segment can lead to worse processing results, for example, for SVM and LR. The performance of strong models is negatively affected by incomplete data for ANFIS, data-deficient and suboptimal choice of architecture for ANN and CNN, and probable overfitting or incorrect setting of parameters on small datasets for XGBoost. Replacing the model on the segment with another model with better performance is possible in such situations. Nevertheless, the methodology shows the possibility of improving the MSE and MAE performance for relatively simple models in information sequence segmentation by selecting segments and assigning models to them.

5- Discussion

This paper proposes a methodology for analyzing the quality metrics of the processing models. In contrast to known solutions that perform the tuning of machine learning models to the processed data, this new methodology divides a sequence into segments so that the properties of the data obtained in the subsequence best match the processing model. Therefore, the model is tuned on a data sample and subsequences with data properties are formed and selected to fit the

processing model. Such an approach generates subsequences from a data sample and then tests machine learning algorithms and processing models on them. The selection criterion was designed to achieve the best quality factor when all possible variants were attempted, allowing the selection of the best available model.

Reducing the size of the segments by increasing their number is reasonable up to a certain point. There was no significant improvement in the quality of the data processing models after reaching the limit. Assigning algorithms with better quality indicators to the segments improves the quality indicators of sample processing compared to individual classifiers and group methods by 3-8%.

The results show that segmentation has a greater impact on improving the performance of linear classifiers. The impact was smaller in the case of nonlinear classifiers. Table 4 provides an overview of forecasting on various datasets using and without the proposed solution. Segmentation consistently leads to improvements for all baseline methods on all datasets, with an average improvement in MSE and MAE of approximately 8% for weak models and approximately 3% for strong models. The proposed technique allows preparing univariate and multivariate data for weak models LR, SVM, GR models for different datasets was 5%-8%, for stronger models DT, CNN, ANN, ANFIS, XGBoost 1%-4%. For strong models, the technique improves results for samples with small data. In both cases, the technique allows segmenting samples of information sequences on which the segment data agrees better with the training models.

Although direct comparisons between the results presented in this study are difficult due to the different conditions and approaches used for data processing, an indirect assessment was performed by comparing the calculated quality indicators with other works using different approaches to data sample segmentation and separation.

Implementation of the graph model [40] for finding break points for dividing time series improved data processing quality by 8%. The use of a fuzzy mechanism for identifying break points of regression sequences [40] improved the quality of the model processing by 7%. Data clustering, as studied in Piernik & Morzy [49], improved the model's performance for classification problems when dividing a sample using class labels and a distance function within 3%-5%. The use of the author's "Segment, Shuffle, and Stitch" approach [50] for data sequences using neural network processing increased the quality indicators by an average of 5% for various datasets. The integration of regression models performing preprocessing and data division into the deep learning models proposed in Shmuel et al. [51] increased the MSE indicator from 4% to 11%. Based on the comparison, the proposed method demonstrates acceptable and sometimes superior performance compared to the recently proposed machine and deep learning models. However, direct comparisons are difficult due to differences in the data used to conduct the studies.

The results obtained are comparable with those obtained in previous studies [40, 43, 49-52]. However, the proposed methodology is relatively simple and allows us to divide sequences into different regression data types. Therefore, it is possible to form modular computational structures of various complexity and replace one model with another. Experiments show that segmenting the data sample and applying different approximation models provide a certain level of quality indicator values.

In some cases, the proposed methodology can be used as an alternative to the data augmentation strategy, especially on large datasets. Data augmentation methods are generally more suitable for small datasets. Due to the lack of variability and diversity, their impact is often less pronounced on large datasets [51]. Segmentation allows the use of local properties of segments.

However, the proposed building regression model methodology for data processing has certain limitations. First, the main limitation, as well as many other ML methods, is the problem of sparse data. When applying the proposed solution, the data should replicate the general population's properties to build an adequate processing model. Second, the sample under consideration must have the characteristics of the general population. If this property is not met, the proposed data processing model's adequacy becomes problematic. Third, in the process of dividing the data into segments, situations may appear when there are too few observation objects in the training sample, which does not allow us to accurately estimate the properties and train the model on these segments. Fourth, when using the proposed solution, the data should be analyzed to avoid overfitting.

6- Conclusion

This study presents a methodology for constructing adaptive models of modular ML structures to improve the processing quality of information sequences. Unlike traditional approaches that adapt models to the properties of the entire dataset, this methodology solves the inverse problem of forming and selecting data segments that correspond to pre-selected processing models. The proposed solution enables processing by assigning models with the best selected quality indicators to each data segment. The main limitations of this study include the need to optimize model assignment to overcome computational complexity, the sensitivity of algorithms to segment properties, and issues related to data quality and training samples. In addition, the segment size should be carefully controlled to maintain model adequacy. From a theoretical perspective, this methodology enables the construction of modular models based on segments, providing flexibility and adaptability to various types of data. This approach supports the creation of complex models, where each segment is processed by a specialized function with configurable parameters. This modularity helps

overcome the limitations of existing methods, including sensitivity to boundaries in piecewise regression, susceptibility to noise in XGBoost, and high resource requirements. For simple datasets, the methodology improves interpretability, while for complex problems, it enables the creation of custom computational structures, improving performance, and reducing system limitations.

Future research should focus on optimizing segmentation methods and refining model assignment strategies for high-dimensional data to overcome these shortcomings. Improving the methodology's adaptability to dynamic datasets will increase its practical value. Additional research could explore its integration into zero- and low-precision learning, where automatic segmentation and structure detection improve feature selection and classification efficiency [53]. The methodology is applicable to a variety of domains, including economics, healthcare, and manufacturing, particularly for time series and complex data stream regression analysis. In practice, it can be used to improve productivity, analyze device or system health, and optimize customer service processes. Its advantage is the reduction of repetitive operations in data sequence analysis, which mitigates costs and frees up computing resources. Additionally, the methodology enables the development of self-learning systems that can adapt over time without significant retraining. This makes it highly relevant in environments that require flexible and scalable ML models. Overall, the proposed solution facilitates the development of modular, interpretable, and resource-efficient modeling strategies, which have far-reaching implications for ML and DDM.

7- Declarations

7-1- Author Contributions

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

7-2- Data Availability Statement

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

7-3- Funding

This research is supported by RSF (grant № 25-21-00269, https://rscf.ru/project/25-21-00269/).

7-4- Institutional Review Board Statement

Not applicable.

7-5- Informed Consent Statement

Not applicable.

7-6- 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.

8- References

[1] He, Y., Zhang, X., Kong, X., Yao, L., & Song, Z. (2025). Causality-driven sequence segmentation assisted soft sensing for multiphase industrial processes. Neurocomputing, 631, 129612. doi:10.1016/j.neucom.2025.129612.

[2] Schober, P., & Vetter, T. R. (2021). Segmented Regression in an Interrupted Time Series Study Design. Anesthesia and Analgesia, 132(3), 696–697. doi:10.1213/ANE.0000000000005269.

[3] Pan, Y., Zhang, R., Guo, J., Peng, S., Wu, F., Yuan, K., Gao, Y., Lan, S., Chen, R., Li, L., Hu, X., Du, Z., Zhang, Z., Zhang, X., Li, W., Guo, Q., & Chen, Y. (2025). Morphology generalizable reinforcement learning via multi-level graph features. Neurocomputing, 631, 129644. doi:10.1016/j.neucom.2025.129644.

[4] Britzger, D. (2022). The Linear Template Fit. The European Physical Journal C, 82(8). doi:10.1140/epjc/s10052-022-10581-w.

[5] Jarantow, S. W., Pisors, E. D., & Chiu, M. L. (2023). Introduction to the Use of Linear and Nonlinear Regression Analysis in Quantitative Biological Assays. Current Protocols, 3(6), 801. doi:10.1002/cpz1.801.

[6] Goepp, V., Bouaziz, O., & Nuel, G. (2025). Spline regression with automatic knot selection. Computational Statistics & Data Analysis, 202, 108043. doi:10.1016/j.csda.2024.108043.

[7] Taye, M. M. (2023). Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. Computation, 11(3), 52. doi:10.3390/computation11030052.

[8] Motamedi, B., & Villányi, B. (2025). A predictive analytics approach with Bayesian-optimized gentle boosting ensemble models for diabetes diagnosis. Computer Methods and Programs in Biomedicine Update, 7, 100184. doi:10.1016/j.cmpbup.2025.100184.

[9] Beiu, V., & Zawadzki, A. (2005). On Kolmogorov's superpositions: novel gates and circuits for nanoelectronics? Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., 2, 651–656. doi:10.1109/ijcnn.2005.1555908.

[10] Girosi, F., & Poggio, T. (1989). Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant. Neural Computation, 1(4), 465–469. doi:10.1162/neco.1989.1.4.465.

[11] Marques, H. O., Swersky, L., Sander, J., Campello, R. J. G. B., & Zimek, A. (2023). On the evaluation of outlier detection and one-class classification: a comparative study of algorithms, model selection, and ensembles. Data Mining and Knowledge Discovery, 37(4), 1473–1517. doi:10.1007/s10618-023-00931-x.

[12] Pillonetto, G., Aravkin, A., Gedon, D., Ljung, L., Ribeiro, A. H., & Schön, T. B. (2025). Deep networks for system identification: A survey. Automatica, 171, 111907. doi:10.1016/j.automatica.2024.111907.

[13] Rinaldo, A., Wang, D., Wen, Q., Willett, R., & Yu, Y. (2021). Localizing Changes in High-Dimensional Regression Models. Proceedings of Machine Learning Research, 130, 2089–2097. doi:10.48550/arXiv.2010.10410.

[14] Aue, A., Rice, G., & Sönmez, O. (2018). Detecting and dating structural breaks in functional data without dimension reduction. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 80(3), 509–529. doi:10.1111/rssb.12257.

[15] Jin, Y., & Yang, Y. (2025). A survey on knowledge graph-based click-through rate prediction. Expert Systems with Applications, 281, 127501. doi:10.1016/j.eswa.2025.127501.

[16] Belitser, E., & Ghosal, S. (2025). Bayesian uncertainty quantification and structure detection for multiple change points models. Bernoulli, 31(2), 1181–1205. doi:10.3150/24-BEJ1766.

[17] Haynes, K., Fearnhead, P., & Eckley, I. A. (2017). A computationally efficient nonparametric approach for changepoint detection. Statistics and Computing, 27(5), 1293–1305. doi:10.1007/s11222-016-9687-5.

[18] Yoo, N. Y., Lee, H., & Cha, J. H. (2025). Development of a new general class of bivariate distributions based on reversed hazard rate order. Computational Statistics & Data Analysis, 204, 108106. doi:10.1016/j.csda.2024.108106.

[19] Wei, Y., Du, M., Li, T., Zheng, X., & Ji, C. (2024). Feature-fused residual network for time series classification. Journal of King Saud University - Computer and Information Sciences, 36(10), 102227. doi:10.1016/j.jksuci.2024.102227.

[20] Golzari Oskouei, A., Samadi, N., Khezri, S., Najafi Moghaddam, A., Babaei, H., Hamini, K., Fath Nojavan, S., Bouyer, A., & Arasteh, B. (2025). Feature-weighted fuzzy clustering methods: An experimental review. Neurocomputing, 619, 129176. doi:10.1016/j.neucom.2024.129176.

[21] Bardwell, L., & Fearnhead, P. (2017). Bayesian detection of abnormal segments in multiple time series. Bayesian Analysis, 12(1), 193–218. doi:10.1214/16-BA998.

[22] Wang, Y., Rosli, M. M., Musa, N., & Wang, L. (2024). Improving clustering-based and adaptive position-aware interpolation oversampling for imbalanced data classification. Journal of King Saud University - Computer and Information Sciences, 36(10), 102253. doi:10.1016/j.jksuci.2024.102253.

[23] Huang, J., Chen, P., Lu, L., Deng, Y., & Zou, Q. (2023). WCDForest: a weighted cascade deep forest model toward the classification tasks. Applied Intelligence, 53(23), 29169–29182. doi:10.1007/s10489-023-04794-z.

[24] Lu, K. P., & Chang, S. T. (2019). Fuzzy maximum likelihood change-point algorithms for identifying the time of shifts in process data. Neural Computing and Applications, 31(7), 2431–2446. doi:10.1007/s00521-017-3200-8.

[25] Tallman, E., & West, M. (2024). Bayesian predictive decision synthesis. Journal of the Royal Statistical Society. Series B: Statistical Methodology, 86(2), 340–363. doi:10.1093/jrsssb/qkad109.

[26] Mehta, S., Kumar, L., Misra, S., Patnaik, K. S., & Singh, V. (2025). Nested deep learning with learned network embeddings for software defect prediction. Applied Soft Computing, 174, 113057. doi:10.1016/j.asoc.2025.113057.

[27] Korkas, K. K., & Fryzlewicz, P. (2017). Multiple change-point detection for non-stationary time series using wild binary segmentation. Statistica Sinica, 27(1), 287–311. doi:10.5705/ss.202015.0262.

[28] Wang, Y., Wang, J., Zhang, H., & Song, J. (2025). Bridging prediction and decision: Advances and challenges in data-driven optimization. Nexus, 2(1), 100057. doi:10.1016/j.ynexs.2025.100057.

[29] Li, D., Yang, J., Kreis, K., Torralba, A., & Fidler, S. (2021). Semantic Segmentation with Generative Models: Semi-Supervised Learning and Strong Out-of-Domain Generalization. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8296–8307. doi:10.1109/cvpr46437.2021.00820.

[30] Doan, Q. H., Mai, S.-H., Do, Q. T., & Thai, D.-K. (2022). A cluster-based data splitting method for small sample and class imbalance problems in impact damage classification. Applied Soft Computing, 120, 108628. doi:10.1016/j.asoc.2022.108628.

[31] Stevanović, S., Dashti, H., Milošević, M., Al-Yakoob, S., & Stevanović, D. (2024). Comparison of ANN and XGBoost surrogate models trained on small numbers of building energy simulations. PLoS ONE, 19(10), 312573. doi:10.1371/journal.pone.0312573.

[32] Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University - Computer and Information Sciences, 35(2), 757–774. doi:10.1016/j.jksuci.2023.01.014.

[33] Zhong, Y., Ren, Y., Cao, G., Li, F., & Qi, H. (2025). Optimal starting point for time series forecasting. Expert Systems with Applications, 273, 126798. doi:10.1016/j.eswa.2025.126798.

[34] Lebedev, I. S., & Sukhoparov, M. E. (2024). Improving the Quality Indicators of Multilevel Data Sampling Processing Models Based on Unsupervised Clustering. Emerging Science Journal, 8(1), 355–371. doi:10.28991/ESJ-2024-08-01-025.

[35] Nguyen, D. D., Tiep, N. V., Bui, Q. A. T., Van Le, H., Prakash, I., Costache, R., Pandey, M., & Pham, B. T. (2025). Landslide Susceptibility Mapping Using RBFN-Based Ensemble Machine Learning Models. CMES - Computer Modeling in Engineering & Sciences, 142(1), 467–500. doi:10.32604/cmes.2024.056576.

[36] Xu, S., Liu, J., Huang, X., Li, C., Chen, Z., & Tai, Y. (2024). Minutely multi-step irradiance forecasting based on all-sky images using LSTM-InformerStack hybrid model with dual feature enhancement. Renewable Energy, 224, 120135. doi:10.1016/j.renene.2024.120135.

[37] Wang, C., Li, X., Zhou, T., & Cai, Z. (2024). Unsupervised Time Series Segmentation: A Survey on Recent Advances. Computers, Materials & Continua, 80(2), 2657–2673. doi:10.32604/cmc.2024.054061.

[38] Kerr, D. J., & Yang, L. (2024). Viewpoint: prognostic markers can personalize adjuvant therapeutic options. Academia Oncology, 1(2). doi:10.20935/acadonco7393.

[39] Lebedev, I. S., & Sukhoparov, M. E. (2023). Adaptive Learning and Integrated Use of Information Flow Forecasting Methods. Emerging Science Journal, 7(3), 704–723. doi:10.28991/ESJ-2023-07-03-03.

[40] Liu, C., Kowal, D. R., Doss-Gollin, J., & Vannucci, M. (2025). Bayesian functional graphical models with change-point detection. Computational Statistics & Data Analysis, 206, 108122. doi:10.1016/j.csda.2024.108122.

[41] Osipov, V., Nikiforov, V., Zhukova, N., & Miloserdov, D. (2020). Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers. Neural Computing and Applications, 32(18), 14885–14897. doi:10.1007/s00521-020-04843-5.

[42] Semenov, V. K. (2025). Sequence-segmentation. GitHub, San Francisco, United States. Available online: https://github.com/vksemenov/sequence-segmentation (accessed on September 2025).

[43] Lu, K. P., & Chang, S. T. (2023). An Advanced Segmentation Approach to Piecewise Regression Models. Mathematics, 11(24), 4959. doi:10.3390/math11244959.

[44] Xhu, X. (2010). Stream Data Mining Repository. Florida Atlantic University, Boca Raton, United States. Available online: http://www.cse.fau.edu/~xqzhu/stream.html (accessed on September 2025).

[45] MathWorks (2025). Regression, The MathWorks, Natick, United States. Available online: https://www.mathworks.com/help/stats/regression-and-anova.html?s_tid=CRUX_lftnav (accessed on September 2025).

[46] Kaggle. (2017). E-Commerce Data. Kaggle, Mountain View, United States. Available online: https://www.kaggle.com/datasets/carrie1/ecommerce-data (accessed on September 2025).

[47] Kaggle. (2019). Hourly energy demand generation and weather. Kaggle, Mountain View, United States. Available online: https://www.kaggle.com/nicholasjhana/energy-consumption-generation-prices-and-weather/data?select=energy_dataset.csv (accessed on September 2025).

[48] Kaggle. (2016). Pima Indians Diabetes Database. Kaggle, Mountain View, United States. Available online: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database (accessed on September 2025).

[49] Piernik, M., & Morzy, T. (2021). A study on using data clustering for feature extraction to improve the quality of classification. Knowledge and Information Systems, 63(7), 1771–1805. doi:10.1007/s10115-021-01572-6.

[50] Etemad, A., Grover, S., & Jalali, A. (2024). Segment, Shuffle, and Stitch: A Simple Layer for Improving Time-Series Representations. Advances in Neural Information Processing Systems 37, 4878–4905. doi:10.52202/079017-0158.

[51] Shmuel, A., Glickman, O., & Lazebnik, T. (2024). Symbolic regression as a feature engineering method for machine and deep learning regression tasks. Machine Learning: Science and Technology, 5(2), 25065. doi:10.1088/2632-2153/ad513a.

[52] Otelbaev, M., Durmagambetov, A. A., & Seitkulov, Y. N. (2008). Conditions for existence of a global strong solution to one class of nonlinear evolution equations in Hilbert space. Siberian Mathematical Journal, 49(3), 498–511. doi:10.1007/s11202-008-0048-2.

[53] Petukhov, I., Steshina, L., & Tanryverdiev, I. (2014). Remote Sensing of Forest Stand Parameters for Automated Selection of Trees in Real-Time Mode in the Process of Selective Cutting. 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, 390–395. doi:10.1109/uic-atc-scalcom.2014.113.