ISSN 0027-1349, Moscow University Physics Bulletin, 2024, Vol. 79, Suppl. 2, pp. S872–S889. © Allerton Press, Inc., 2024.

Feature Selection Methods for Deep Learning Models of Soft Sensors in Oil Refining

I. S. Lazukhin[1]*, M. I. Petrovskiy[1], and I. V. Mashechkin[1]*****

[1]*Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Moscow, Russia*

Received October 1, 2024; revised October 10, 2024; accepted October 20, 2024

Abstract—The development of automated control systems results into industrial plants accumulating large amounts of data on the continuous state of technological processes. Multiple physical sensors record the system states at any given time, hence being crucially responsible for controlling the system and maintaining its parameters within hard limits. At the same time, irregularly conducted laboratory measures make up a significant part of the qualitative indicators of such processes, especially in the petrochemical industry. Mathematical models that generalize laboratory measured indicators to match the frequency of physical sensors are called soft sensors. On practice, soft sensors for laboratory data are represented by linear or last-recorded-value models. We investigate the task of analytically obtaining chemical indicators of the technological process in real time based on the values of physical sensors; the study is conducted on a real-world data set. Several problems are covered, including high dimension of the physical inputs compared to the laboratory data volume; scarcity of the laboratory data collected on a daily basis. Authors propose feature selection methods based on PLS regression (hierarchical clustering), Bayes trees, utilize existing graph neural network, as well as compare developed methods with existing popular approaches. Each of the proposed feature selection methods has been adapted to take into account the expert opinion of the industrial plant engineers. Authors investigate developed approaches alongside neural network methods for predicting time series including graph neural networks, fully connected and recurrent networks. The obtained experimental results show the advantage of using proposed feature selection based on PLS and Bayes in ensemble with simple recurrent networks or graph neural networks with preliminary interpolation. Separately, it is worth noting the ambiguity of assessing the developed models quality; authors propose a combined approach that takes into account the adequacy of the model, its correlation with the true laboratory values and averaged errors.

Keywords: soft sensors, feature selection, deep learning, graph neural networks

DOI: 10.3103/S0027134924702333

1. INTRODUCTION

Modern production plants provide large amounts if continuous precise data on the states of ongoing technological processes. At the same time, in practice, modern process control utilizes either models representing the process using systems of differential equations, or fairly simple (often linear), partially heuristic, control models that are embedded in the process control proprietary software. Both the first and second types of models are criticized by the experts of the field for their lack of accuracy, which is a consequence of the simplicity and versatility of such models. The ongoing movement towards the digitization of production plants makes exploring machine learning methods for solving the problems of forecasting technological process states relevant [1]. The forecast models are built on a case-to-case basis, utilizing historical data collected on particular industrial installations. We will be focusing on the task of forecasting laboratory indicators of a petrochemical technological process; to be more exact, on the subtask of response-based feature selection for handling the real-world data set under study, rich with highly correlated physical variables.

Physical data of the technological process includes the automated readings of the physical sensors across the system recorded online in regular intervals, usually seconds. Despite having missing values or

*E-mail: ivanlazuhin@mail.ru
**E-mail: michael@cs.msu.ru
***E-mail: mash@cs.msu.ru

outliers, physical sensor readings are aggregated easily into continuous data sets of a suitable time scale for analysis. On the other hand, laboratory indicators cannot be measured by a physical sensor and must be obtained manually by qualified personnel from product samples. This means that each laboratory measuring needs human time and effort resulting into not only large intervals between measures but also involvement of the human factor. Such problems as reusing of the old product samples, measurement mistakes or delays could produce unreliable results.

The laboratory data possesses the following (but is not limited to) features: is very scarce data that cannot be obtained exactly at any given moment in time (timestamp assigned to a measurement corresponds to the moment a product sample is taken); a critical component of the process, represents product quality indicators; cannot be evaluated precisely, each laboratory indicator has high and low limits for reproducing the measurement depending either on the sample itself or on the laboratory conditions. We will define a soft sensor as an approximation model for the laboratory data of a fixed parameter measurement. Most popular interpolations used in real production plants are linear and piecewise constant approximations.

2. OVERVIEW OF THE EXISTING SOLUTIONS

The development of soft sensors increases the production plant efficiency across the modern industry. Multiple academic studies were considered below, although the area of study contains a lot of commercial products with closed and/or limited to a single plant implementations.

A study [2] investigates feature selection methods for an oil distillation real-world data set, including approaches based on information criteria, correlation analysis and others. The study concludes that none of the provided methods was able to improve the pure expert knowledge-based models, moreover, all the methods discarded the variables selected by technicians. The proposed models implement a combined approach: feature selection models rank variables provided by experts. At the same time, the data under study contains relatively small amount of input features available for close expert inspection.

A more recent [3] focuses on the similar problem of approximating scarce laboratory measures (recorded from a real wastewater treatment plant). While not providing significant innovations in the field of soft sensoring (proposing ridge regression as the best model), the authors conclude that the black box approaches show less efficient results than those incorporating the knowledge of the plant structure.

A more typical (for the task of constructing soft sensors) approach utilizing fully connected autoencoders is discussed in [4]. The study is focused on modelling multiple quality indicators of the petroleum refining process and proposes using quality-driven regularization for the task of implicit feature selection. The experiments show that response-based feature selection has advantages over the simple solution of building auto-encoders on the basis of abundant physical data. Another earlier [5] discusses autoencoders for working with missing data, although the degree of missing values under study (maximum is 50%) is far from the one discussed in our study (99%).

In the work [6], a generative approach is used for augmenting the laboratory readings data set. The rCWGAN architecture (conditional generative-adversal neural network with regression) is proposed, integrating laboratory readings as condition variables into the model training. The study focuses on modelling the purified terephthalic acid (PTA) solvent system. We will note that the generative approaches show low efficiency at the moment on the data studied in our work.

The works described above fail to represent the task of modelling laboratory data as a task of approximation for the limited set of points in time. Most studies simply represent it as a straightforward regression between a set of physical variables and a lab data measure in a fixed moment in time. At the same time, the technological process under study has a significant delay before any physical sensor value will have its impact on the product. Another problem not accounted for in the described works is critically small size of the data set. Compared to the mentioned in the overview processes, the physical process under study is hardly interpretable with a single diagram, containing hundreds of physical sensors. The overview raises the following relevant points of developing soft censors: utilizing expert (process) knowledge while not focusing on the physical understanding of the process; handling large amounts of physical variables with regard for the response; modelling laboratory data with time dependences. The last problem we will address in this section is the lack of open datasets for studying in the applied field.

3. INDUSTRIAL DATA AS MULTIDIMENSIONAL TIME SERIES

Here, we will discuss the problem of representing both physical and laboratory data in terms of time series and introduce the main tasks of recording and modelling the industrial process.

3.1. The Task of Forecasting Industrial Time Series

To develop a model of a process in time we firstly need to define the time scale and a way to translate sensor data into a form suitable for processing. Here, we will be focusing on discrete time models, representing the system like a finite state machine with memory, contained in the sliding window of a predefined size. Let us introduce a fixed discrete grid in time:

$$T = t_1, t_2, \ldots, t_M, \quad t_{i+1} - t_i = \tau = \text{const}.$$

Assuming a subset of the studied real physical sensors of the plant as continuous functions of time f_i, we can construct the following projection:

$$F(t) = f_1(t), \ldots, f_N(t),$$
$$A := [f_j(t_i)]_{1 \le i \le M, 1 \le j \le N}.$$

Thus, the matrix of sensor readings A discretely displays the production process over the period of time $t_1 \ldots t_M$ with a uniform frequency of τ.

Let us fix a laboratory indicator lab_i and adjust the one-dimensional set of obtained measures of size m_i onto a grid corresponding to the physical indicators (the method of reduction is based on choosing the closest timestamp):

$$\text{lab}_i(t) : t \in t_{k_1} < t_{k_2} \ldots t_{m_i - 1} < t_{m_i} \subset T.$$

By projecting a set of laboratory readings onto the discrete grid of physical indicators of the process T, we obtain a sparse (by missing values) matrix of readings:

$$]B := [\text{lab}_j(t_i)]_{1 \le i \le M, 1 \le j \le P}, \quad AB := A \text{ join } B.$$

This is the form that will further contain the technological process data as an input to the developed algorithms. The matrix AB will represent a multidimensional time series or a data set. Here and below, we will omit the details of transforming the raw readings of physical sensors to the resulting grid of specified fixed frequency τ.

We will build mathematical models of laboratory variables in discrete time with physical indications as inputs. Let us omit mentioning of undefined values further on, assuming by default that they do not participate in the machine learning methods (feature and response vectors containing missing values will be discarded), unless explicitly specified (for example, in the case of interpolation).

The time series for the physical sensors A can be represented as a sequence of records A_t, where $A_t = a_{t1}, \ldots, a_{t_N}$ (likewise, $B_t = b_{t1}, \ldots, b_{t_P}$). Then, the task of predicting the state $t + k$ ($k > 0$—step of the model), from the sliding window of history values sized r (correspondingly, lag of the model) up to the moment t included could be solved by constructing individual models for the necessary step in time:

$$B_{t+k} \approx \hat{B}_k(A_t, \ldots, A_{t-r}), \quad t > k.$$

Thus, arises the task of selecting the optimal type of the desired regression function $\hat{B}$ for modeling laboratory indicators of technological processes. We will focus on selecting such a target function among regressions based on neural network methods. Taking into account the specifics of the task, a variety of methods were selected that correspond to the model described above.

To summarize this section, the main task of this paper is to simulate laboratory readings (or to construct the soft sensors) using historical readings of physical variables.

Table 1. Numerical characteristics of the data set

Parameter	Value
Physical data measures	8569
Physical sensors	322
Laboratory measures	326
Physical sensor frequency	1 h

3.2. Real World Physical Sensors and Laboratory Measures Data Set

This work focuses on a data set,[1] collected at an oil refining facility in a span of a year. The data set includes a large set of physical variables, as well as one target laboratory indicator (the target variable corresponding to the final boiling point of the product) selected by experts for modelling. Due to the rarity of such data sets and studies on them in the public domain (especially highly multidimensional data sets containing sparse laboratory readings), many decisions made during this study relied on experts of the field and the specifics of the data sets available to the authors of the study.

The target variable has an average and median periodicity of 25 and 24 h, accordingly. More detailed characteristics of the studied set can be seen in Table 1. The distribution of the raw laboratory readings of the target variable is shown in Fig. 1. The monthly volume of target variable readings can bee seen in Fig. 2. The target variable plot is present in Fig. 3.

[1] A data set provided to the Lomonosov Moscow State University by the PJSC Lukoil Oil Company as a part of the corresponding research contract no. ITS 1-22-26sp dated October 16, 2023.

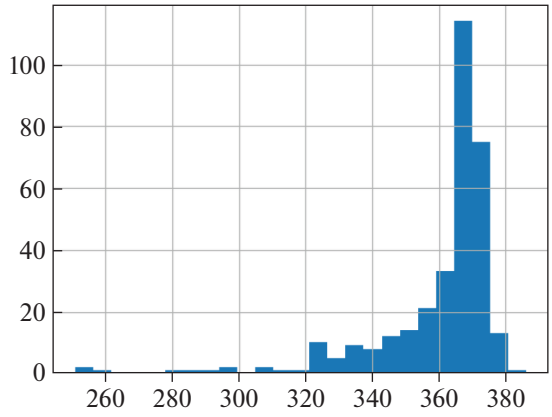

Fig. 1. Target variable distribution.

We highlight the following problems of the studied data: a relatively small amount of readings for the effective construction of complex statistical models; presence of outliers (fictitious data as well) in the readings due to both external influences and human factor; presence of several different periods of the plant operational mode in the data set (see Fig. 4), including periods not suitable for research. The target variable has such attributes as reproducibility and repeatability, coming from the applied field and determining the accuracy of measurement in different laboratories or in different experiments accordingly. We will take into account the repeatability threshold when evaluating models.

The number of laboratory test readings closely coincides with the the number of physical variables; at the same time, the data set contains a lot of strongly correlating physical variables (see Fig. 5), therefore, we will mostly focus on the selection of the most significant physical variables as a way of improving models. This is further justified by the need of the applied field experts to participate in the construction of the soft sensors.

4. FORECASTING LABORATORY INDICATORS

The following soft sensor modelling subtasks were covered in this work:

1. Feature selection, in order to reduce the dimension of the feature space—the volume of readings of the target variable is close or marginally lower than the dimension of the input feature space.

2. Filling in missing values (points of time when the laboratory measures weren't conducted) of the target variable in order to increase the volume of data suitable for training models and expand the classes of applicable forecasting algorithms.

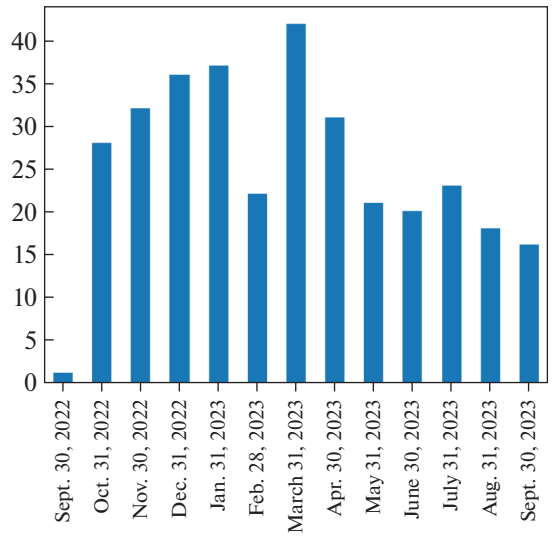

Fig. 2. Monthly volume of target variable measures.

3. Forecasting of laboratory readings based on sliding windows of historical data of physical variables as features.

4.1. Basic Preprocessing

Stable periods of plant operation based on simple heuristics and expert knowledge of the process were semiautomatically selected—restrictions on the derivatives and the absolute values of the the target variable readings were considered for the task. Final approach implies filter limiting weekly difference between averaged laboratory measures. Further mentions of the lab indicators data set include this procedure as basic preprocessing. A simple standard scaling was used to bring both features and the target variable to a similar scale. Taking the discrete derivatives of the physical variables hadn't shown significant difference on the models accuracy.

Since some of the machine learning approaches described further require continuous data for correct processing, several basic interpolation tools have been considered to fill in missing values in the laboratory readings. Five base concepts were considered, including spline interpolation (Fig. 6), LOWESS regression [7] (Fig. 7), naive linear and constant interpolations (as the most frequently used in the field), as well as Gaussian kernel regression (Fig. 8). Spline and LOWESS methods were chosen for further consideration via preliminary experiments. A 48 h limit on interpolating gaps was used to prevent unrealistic data.

4.2. Feature Selection

The main problems of feature selection for the task under research (some of which have already

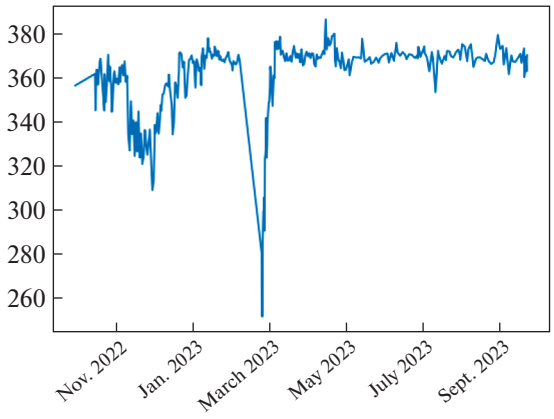

Fig. 3. Plot of the target laboratory variable readings on the period under study.

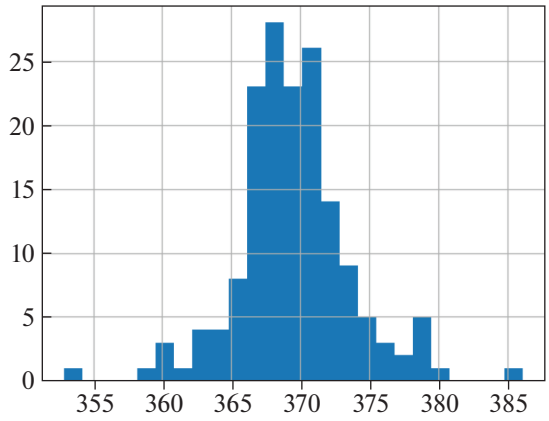

Fig. 4. Target variable distribution (after March, 2023).

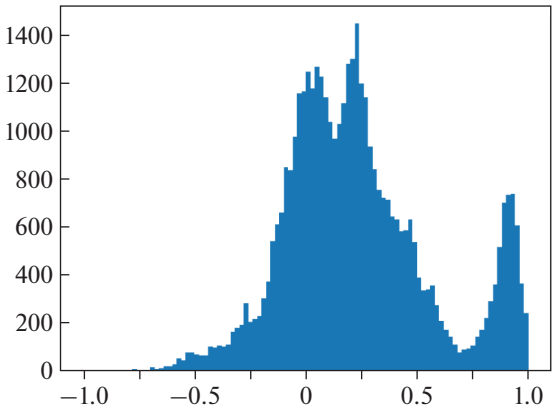

Fig. 5. Distribution of nonzero correlation coefficients between physical variables.

been mentioned when discussing the studied data set above) include the abundance of high correlations among the variables of the feature space, its large dimension relative to the data volume of the target variable, as well as an unknown time delay of the impact individual physical variables have on laboratory studies.

The issue of delayed response between laboratory indicators and physical control actions reflected in the time series of physical variables is investigated separately. After, we propose approaches to reducing the feature space based on several promising classes of algorithms; then we describe the methods of changing the feature space that are popular in the field of time series prediction, adapted for use in the framework of the studied data set.

4.2.1. Time component. Technological processes usually have a noticeable time delay between the time stamp corresponding to the control action and the display of its effect on dependent indicators, individual for each control and dependent variable. Likewise, in the case of laboratory measures, such a delay depends on the rate of change in the chemical parameters of the plant products through control actions. This trait requires additional transformations of the feature space to establish a direct concordance between the target and feature data sets.

For feature selection models that do not explicitly take into account the time dependences between inputs and outputs, we will look for the optimal shift of the feature space along the time axis before selection. Pairwise correlations of the target laboratory variable and each physical feature are calculated for multiple time delays within the fixed range (a day as the average time between receiving laboratory measures). In general case, a variable shifted in time to its maximum correlation with the response according to the fixed criterion is investigated. We propose calculating the Spearman correlation and assess the importance based on the obtained p-values in the ascending order. Let us call this approach an implicit time component—the subsequent feature selection algorithm does not take time into account and processes data where the time component of production has already been taken into account. An example of visualizing the correlation of variables and their shift relative to the moment a laboratory measure is taken is shown in Fig. 9.

An alternate way in case of feature selection models that do not take time into account is to study explicit time dependences between physical and laboratory variables as a feature space. Two-dimensional features are built—each feature is a pair representing $(variable, lag)$; then the feature selection models are fed flattened data. We will call such a time component explicit, also widely known as sliding-window approach.

Models that take into account the time component at the algorithm level (for example, recurrent neural networks) already calculate the temporal dependence,

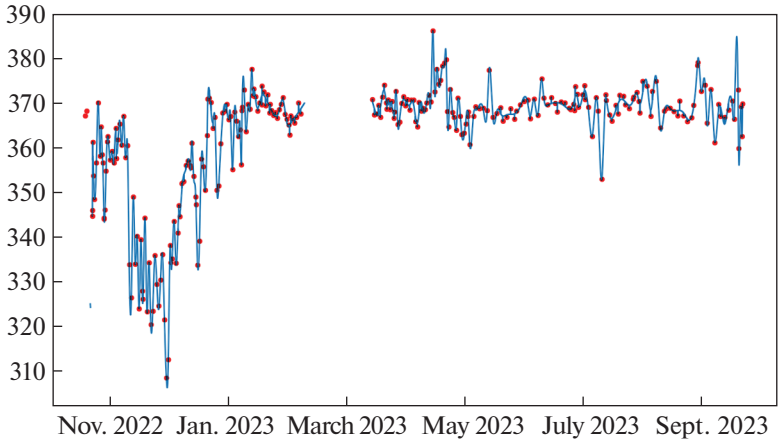

Fig. 6. Spline interpolation, filtered data.

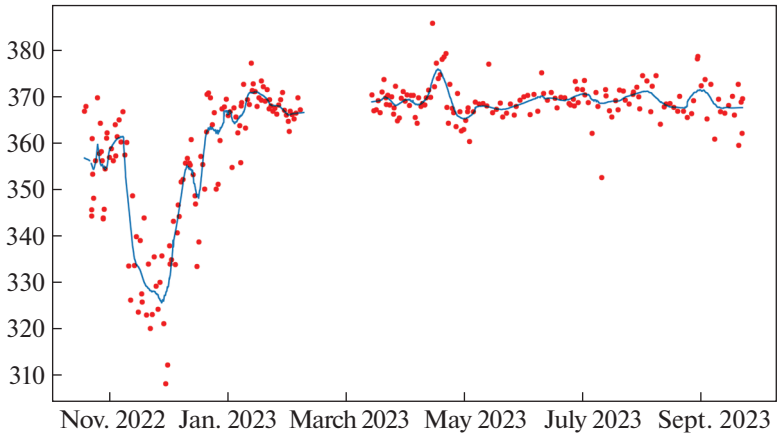

Fig. 7. LOWESS interpolation, filtered data.

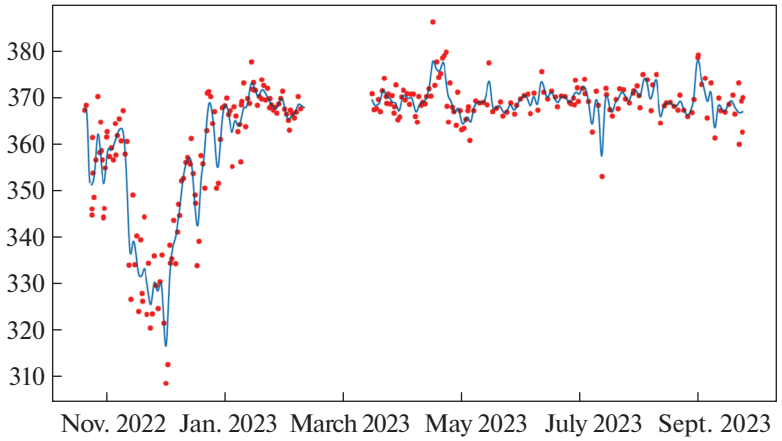

Fig. 8. Gaussian kernel regression, filtered data.

therefore they do not need preprocessing of the feature space. Let us call such a time component automatic.

4.2.2. PLS-clustering. A popular classical method of reducing feature dimensionality is the principal component analysis (PCA), which decomposes the original feature space into components (linear combinations of variables) describing the largest proportion of variance in the absence of other components, intuitively, constructing an n-dimensional ellipsoid on top of the original n-dimensional data.

Similar to PCA method, partial least squares (PLS) is a regression method based on the projection of the inputs and outputs of the model onto a new latent space of a smaller dimension [8]. The PCA algorithm looks for a subspace of inputs corresponding to the direction explaining the largest proportion of output variation. Consider the NIPALS algorithm [8], which we use further to fit the PLS model. Let X and Y correspond to inputs and outputs (columns correspond to features, and rows correspond to observations; let us pay attention to the possible multidimensionality of the outputs, but consider the one-dimensional case for simplicity), a x_i corresponds to the i-column of the matrix X (and similarly for others). We will look for matrices T and U that maximize the covariance (correlation) between X and Y in the space of the found components (matrices P and Q):

$$X = TP^T + E, \quad y = UQ^T + F.$$

Components are found iterative; for each fixed step i of the algorithm we search recursively for the i-component, until we find suitable threshold for u_i:

$$Initialization : E_1 = X, F_1 = y, u_1 = E_1$$
$$E \to U : w_i = normalized(F_i^T u_i \cancel{||u_i||^{-2}})$$
$$E \to W : t_i = E_i w_i$$
$$F \to T : c_i = F_i^T t_i ||t_i||^{-2}$$
$$F \to C : u_i = F_i c_i ||c_i||^{-2}$$

where C comes from the implicit task of regression $X \to Y$. Then we remove variance already explained by E_i and F_i:

$$E \to T : p_i = E_i^t ||t_i||^{-2}$$
$$E_{i+1} = E_i - t_i p_i^T, F_{i+1} = F_i - t_i c_i^T$$
$$u_{i+1} = E_{i+1}$$

After that, the algorithm continues for the $i+1$ component. It is not difficult to show that for a direct transformation it is possible to obtain the rotation matrix R:

$$R = W(P^T W)^T, \quad t_i = XR_i, \quad \hat{Y} = TC. \qquad (1)$$

We propose the application of this method for hierarchical clustering (similar to existing solutions based on PCA [9]) and its adaptation to utilize the expert's opinion in the feature selection process. Initially, the field expert prepares a black list of variables, selects a single target variable or a group. Next, for a set of features, the following approach is performed, represented as a hierarchy or a tree: a PLS regression with two main components is trained on the input set of features, the proportion of variance explained is calculated—by individual variables and by the entire model; variables are divided into subgroups according to the amount of contribution to one of the components. The algorithm continues recursively for groups of variables of the first and second components.

In more detail, for the data set X, on the current cluster of variables, the affiliation of the variable i to the first component is defined as $|R_{i1}| > |R_{i2}|$, where R comes from Eq. (1). Similarly, the proportion of variance explained by the variable i relative to the target variable y (column of Y) will be calculated as:

$$R^2(y|x_i) = 1 - \frac{\Sigma(y - \hat{y})^2}{\Sigma(y - \text{mean}(y))^2},$$
$$\hat{y} = x_i(R_{i1} + R_{i2})C.$$

The algorithm stops when the required splitting depth is reached. The variables are sorted within resulting clusters by the proportion of unexplained variance, and either expert (the first priority) or first-order variables are selected as cluster representatives. In more detail, the following formula is proposed for sorting variables within the I cluster:

$$\text{score}(i) = \frac{1 - R_i^2}{1 - \max\limits_{i \in I} R^2}, \quad i \in I.$$

The resulting tree is visualized as a dendrogram, vertically—the proportion of the explained variance of the constructed models, see Fig. 10. Automatic selection implies the fixed required level required of explained variance or depth as a threshold to stop the tree growth. The full algorithm is presented schematically in Fig. 11. Expert selects initial variables and choses the important variables within found clusters without going into the details behind building the hierarchy.

The proposed approach allows for the selection of features for several target variables simultaneously, takes into account the expert's opinion during data preparation, and also implies interpretation in the form of a dendrogram. The time component is taken into account implicitly, that is, a preliminary shift of features in correlation with the response is necessary. The algorithm has one hyperparameter—the tree pruning threshold.

4.2.3. Auto-generated Bayesian networks. Feature selection based on Bayesian networks was developed as well, and the Chow–Liu method of generating Bayesian trees turned out to be suitable for this paper's problem in terms of computational complexity. The Chow–Liu tree [10], as a kind of Bayesian network that approximates the joint distribution of $P(X_1, X_2, \ldots, X_n)$ in the form of a tree, which is written as a product of conditional probabilities $P(X_k|X_p)$, $1 < k, p < n$, according to the Bayes theorem. The tree implies minimizing the

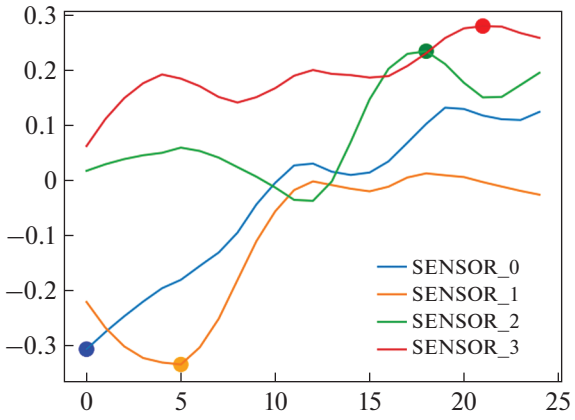

Fig. 9. Visualization of the correlation of several physical variables with the target depending on the lag, the maximum absolute correlations are highlighted by dots.

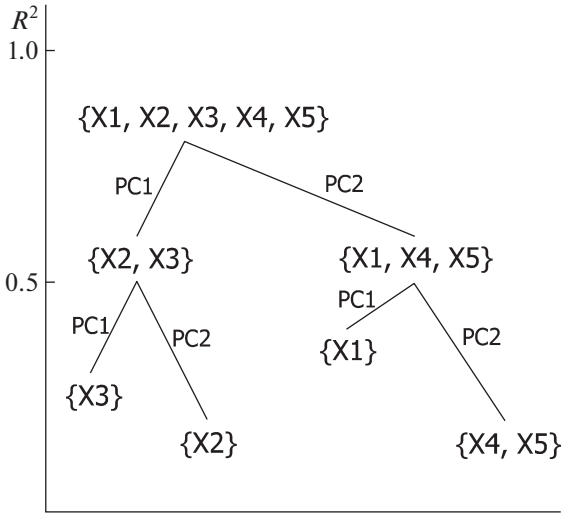

Fig. 10. A schematic process of PLS-clustering represented as a dendrogram.

Kullback–Leibler deviation, which can be written in terms of individual and multiple entropy H as:

$$-\Sigma I(X_i, \text{parent}(X_i)) + \Sigma H(X_i) - \Sigma H(X_1, ..., X_n).$$

The method of constructing the Bayesian tree is simple and involves adding an edge corresponding to the maximum mutual information at each step.

Let us discuss the feature selection we have proposed for the laboratory measures. Initially, the data is reduced to a fixed frequency: the values of laboratory variables are approximated by the values of the nearest timestamp corresponding to the selected frequency. Similar to PLS clustering, the analysis of input and output correlations described earlier is introduced to choose the optimal time series shift for feature selection. Time series (physical and laboratory) are individually sampled by ten (the value could vary for using in different data sets) quantiles

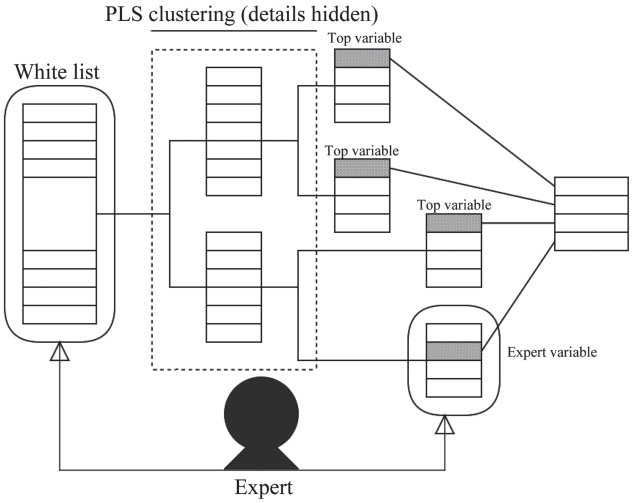

Fig. 11. Expert opinion utilized for feature selection.

based on a predefined training sample. This results into a multiple time series, each consisting of values representing classes for constructing the tree. After that, the shifted discretized data is combined along the time axis and filtered based on missing values (absence of laboratory measures).

For the prepared table of discretized laboratory data, a Chow–Liu tree is constructed with the target variable as its root. An expert black list of variables implies their absence from the source data. As selected features, it is proposed to choose a tree based on the preliminary selected depth or a fixed number of features. The ranking of features is based on their depth. Insignificant branches are cut off according to the chi-square criterion with a significance level of 0.05.

Field expert interaction implies the addition of variables that are required to be included in the graph (a white list of variables at the stage of tree construction)—they are given priority when choosing edges; as well as the construction of directional connections of interest using the expert knowledge (after the tree generation stage). The selection is meant to be individual, since the chosen tree preparation algorithm requires a single root node. Method implies visualization in the form of a tree, a real example can be seen in Fig. 12. The algorithm has two hyperparameters—depth of the tree/number of variables and the p-value threshold for trimming links.

4.2.4. StemGNN. StemGNN is a deep neural network utilizing a spectral-temporal graph and an attention mechanism proposed in the work [11]. The authors of the original architecture note that when predicting multidimensional time series, it is necessary to simultaneously take into account both correlations within time periods (correlations between

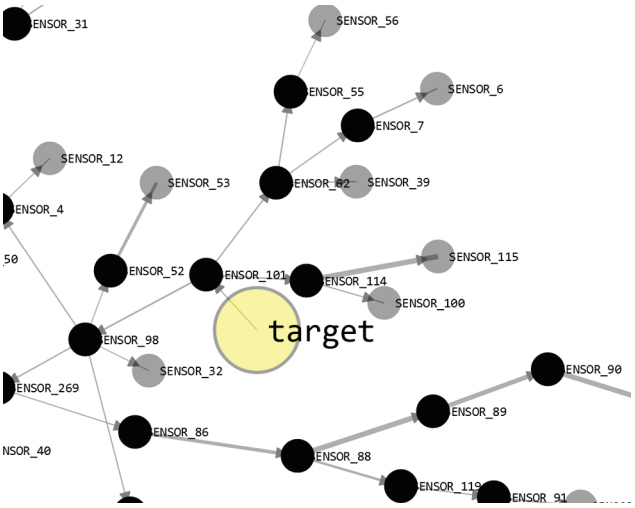

Fig. 12. Feature selection visualization based on the Chow–Liu tree, variables are anonymous.

different features) and correlations between time periods (correlations within the values of a feature), which was taken into account when creating StemGNN through the use of the graph Fourier transform—GFT (takes into account correlations between time periods periods) and discrete Fourier transform—DFT (takes into account correlations within time periods). First, a GFT transformation takes place, which converts the structural multidimensional input data into a set of spectral time series, while the various trends can be decomposed as orthogonal time series. The DFT transformation is used to transfer each individual time series into the frequency space.

After applying these transformations, a spectral representation of the time series is obtained, containing clear patterns that can be effectively predicted using convolution and sequential learning modules. Moreover, the neural network includes a hidden level of automatic analysis of correlations between time periods. Forward and reverse forecasting modules with a common encoder are used to facilitate the representation of time series.

We propose using this neural network architecture for feature selection. In the first StemGNN layer, the attention matrix is automatically created between the features (see Fig. 13), which is used to rank variables representing physical sensors. The laboratory data interpolated by a fixed method is combined with the readings of physical sensors into a common data set and passed through the StemGNN graph neural network model, to generate a matrix of pairwise attention (graph representation of a time series).

A complete predictive model of the united time series is built (the original architecture assumes the prediction of the next vector value of a multidimensional series for a given sliding window-history), after which the first layer corresponding to the attention layer between the variables of the series is extracted from the model. As a hyperparameter, it is necessary to choose the complexity of the StemGNN model, for our task we will fix the simplest option in the form of a single StemGNN block (see Fig. 13).

While this is not a concern for the feature selection, the StemGNN model requires autoregression for forecasting time series which is complicated in the case of constructing soft sensors on the scarce data, even after interpolating. We propose modifying the architecture by adding a dummy input representing the soft sensor as a weighted linear combination of all the input time series. In order to simplify the comparison of architectures, we will use this modification both when selecting features and using StemGNN as an independent predictive model.

The resulting first layer of the model contextually generates attention for a given moment in time in the form of a weighted connectivity matrix. The expert can set a normalized attention weight of 1 for selected variables before/during training (which ensures their inclusion in the model), as well as obtain contextual importance on new data. The feature selection takes place by sorting the averaged attention matrix for a given period of data and a laboratory variable (a fixed desired number of physical variables is required for selection).

StemGNN-based feature selection does not require additional care for the time component, processes several laboratory variables simultaneously and has one hyperparameter in the form of the desired number of selected variables.

Separately, we consider the possibility of extracting the first layer of the model that builds the attention matrix and using it as an attention layer for new neural network models. In this case, the expert will have the opportunity to interact with the feature space online, which will allow taking into account the expert opinion contextually.

4.2.5. Response-based L_2-regularization. In [4], a pseudo approach to feature selection in fully connected networks was proposed, based on a regularization multiplier, depending on the correlation with the response. We propose a modified approach that generalizes regularization to networks of any class and takes into account the expert's opinion, applying response-based regularization to a layer implementing a linear combination of the time series at the input of the sliding window model. The other modification is utilizing this approach to analyse sliding windows in the form of recurrent models.

A correlation vector (of a fixed type) of inputs and laboratory studies is constructed (individually), which

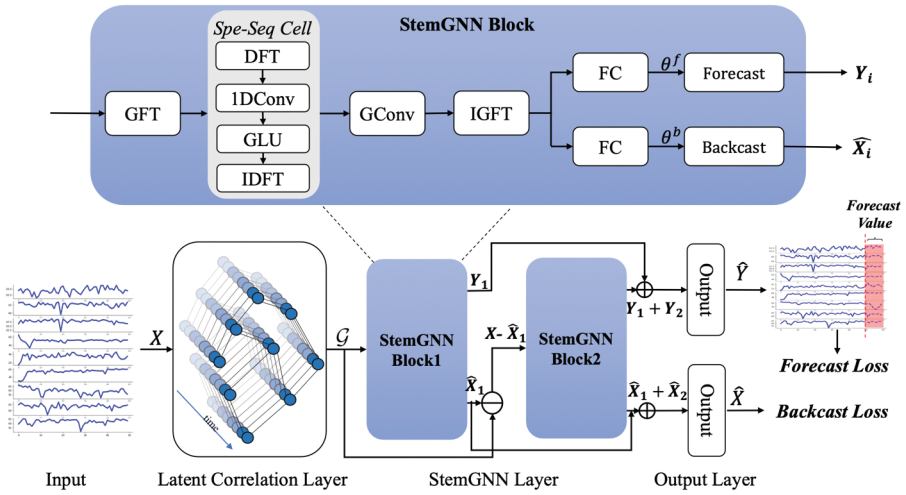

Fig. 13. Scheme of the StemGNN architecture as presented in the original work.

is added as a multiplier for regularization of the interpolation model of laboratory indicators/prediction of soft sensors. The expert puts down maximum values ("1" in case of positive correlation) as multiplier for physics-based necessary dependences.

To summarize, the weights of the model during gradient descent at step t for the input layer (the neuron with the number k is represented) are updated according to the following formula:

$$W_{jk}^{t+1} = W_{jk}^{t} - \frac{\delta(J + a(1-\rho_j)(W_{jk}^{t})^2)}{\delta W_{jk}^{t}},$$

$$W_{jk}^{t+1} = W_{jk}^{t} - (\frac{\delta J}{\delta W_{jk}^{t}} + 2a(1-\rho_j)W_{jk}^{t}),$$

where ρ_j corresponds to the correlation coefficient of the input variable j and the output, J corresponds to the loss function; a corresponds to the basic regularization multiplier for L_2.

The method does not provide the ranking of features explicitly, such can be indirectly considered the rating of Spearman correlation coefficients between the target and physical variables. Hyperparameters include the chosen type of correlation and the base regularization coefficient.

4.2.6. Existing approaches to feature selection. The authors also investigated a set of established feature selection methods as an alternative to the proposed ones and also to expand the classes of tested approaches.

The feature selection method based on decision trees, in particular, gradient boosting LightGBM [12], was investigated in two variants. The first one is the construction of a sliding window model in time from physical variables with one target variable, a laboratory indicator. Such model implies an explicit flat representation of features (an explicit time component) in the form of a lag in time and a variable name. An example of features selected for the data set under study can be seen in the Fig. 14. Alternatively, a model with a preliminary corrected time component was investigated. The selection of variables, or variable-lag pairs, is performed by ranking the number of feature occurrences in the ensemble. The LightGBM parameters that differ from default ones are represented in Table 2.

The LASSO feature selection method [13] with preliminary selection of the time component shows relatively sparse results (based on the coefficient matrix, see Fig. 15) for the problem under research, therefore features are selected by the threshold on the model coefficients.

Separately, recurrent autoencoders were considered as a way to transform the complex data of a sliding window of physical variables to an abstract representation of lower dimensionality. At the same time, autoencoders allow partially solving the problem of small data volume by depending on only continuous data of physical variables when building the models [14]. A simple autoencoder was proposed to use for feature transformation, consisting of two GRU layers [15], one to transform the input sliding

Table 2. LightGBM parameters

Parameter	Value
Estimators	10
Learning rate	0.1
Max depth	4
Max leaves	16

Table 3. The studied feature selection algorithms and the features selected by multiple methods. The variable names in the index column are anonymous and present only to avoid confusion when assessing the table. Numerical values in the cells correspond to the model's provided scoring

Sensor	PLSC	Bayes	LASSO	LGBM	StemGNN
SENSOR_1	0.865768	4	0.045041	3	–
SENSOR_2	–	4	0.029486	–	0.947341
SENSOR_3	0.963523	4	0.123929	–	–
SENSOR_4	–	2	0.034464	–	0.334994
SENSOR_5	0.79364	1	–	3	–
SENSOR_6	–	4	–	2	0.414463
SENSOR_7	0.994848	–	0.032323	–	–
SENSOR_8	0.930119	2	–	–	–
SENSOR_9	–	3	0.137878	–	–
SENSOR_10	0.979286	4	–	–	–
SENSOR_11	–	4	0.203014	–	–
SENSOR_12	–	3	0.021009	–	–
SENSOR_13	–	2	–	–	0.361444
SENSOR_14	–	3	0.001983	–	–
SENSOR_15	–	4	0.05296	–	–
SENSOR_16	–	3	0.017038	–	–

window into the hidden representation of 64 neurons and other to transform the representation back by repeating the representation vector. The model is preliminary fit using optimizer RMSprop [16] with learning rate of 10^{-3} and a batch size of 64.

5. CASE STUDY

Here, we investigate the real world problem of forecasting a selected target variable corresponding to a laboratory indicator of sparse data volume. The data set was previously discussed in Subsection 3.2. One hour was chosen as the minimum forecast horizon, and the interpolation problem is also solved simultaneously with the forecast by generalizing the built forecast model onto physical data. Due to the lack of real data for the missing laboratory values, the forecast metrics will be applied on real readings, and the approximation will be assessed visually as well.

5.1. Forecasting Neural Models

5.1.1. Simple recurrent model. As a standard method for processing sequences, simple recurrent neural networks were analysed for the task; they allow processing sequences with a small amount of parameters, to help preventing overfitting under conditions of critically small data volume.

A simple model consisting of a GRU layer (16 neurons) with batch normalization followed by a fully connected layer at the output was studied.

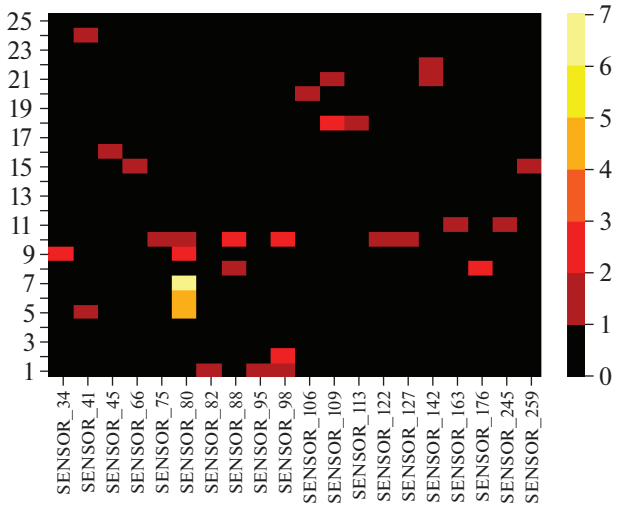

Fig. 14. Two-dimensional features selected using the importance in a tree ensemble.

Over-fitting of the model was prevented by the weight decay (0.01) as well as its simplicity. The Adam optimizer [17] was used with a learning rate parameter of 0.01 and a stochastic gradient descent batch size of 128. Mean squared error (L_2 norm) is used as the loss function.

Recurrent models [18] allow the use of feature selection methods that do not violate the structure of the input data (either transforming the input time series, or selecting variables from the original series). Due to the simplicity of the proposed architectures, they can work with both interpolated data and initial sparse laboratory readings.

5.1.2. StemGNN. The above-discussed StemGNN graph neural model, in addition to feature selection, also provides us with the ability to approximate the continuous series from its input. It is quite complex to be used within a small data set, but the transformation of features into an internal representation by means of varieties of Fourier transformations allows it to be used for a direct (on an interpolated lab data) forecasting task along with simple models.

The model is trained using the RMSProp [16] optimizer (according to the original work), a learning rate of 10^{-5} and a batch size of stochastic gradient descent fixed at 8. Mean squared error is fixed as the loss function.

Like simple recurrent models, StemGNN requires an input structured as a multidimensional time series. Also, we note the impossibility of using the initial noninterpolated laboratory readings in this model due to its great complexity in terms of the number of parameters.

5.1.3. Multilayered perception. Finally, simple fully connected models were investigated in order to compare the proposed structured feature selection methods with methods that transform the original time series space into flat features.

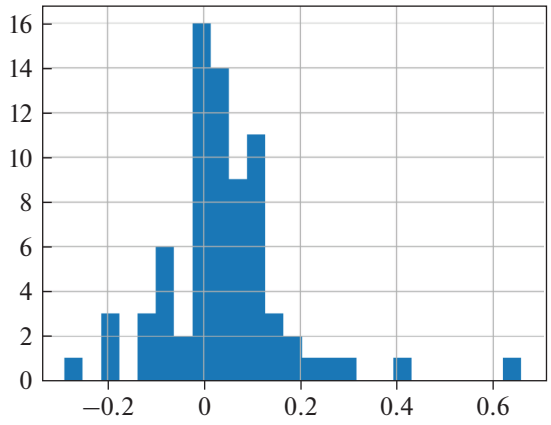

Fig. 15. Nonzero coeficients distribution of the LASSO model.

A simple two-layer model was studied, 128 and 16 neurons, respectively, with the LeakyReLU activation function and the normalization of the batch between the layers. Note that the proposed model is trained using the Adam optimizer, learning rate 0.01, weight regularization 0.001 and the stochastic gradient descent batch size 8. Mean squared error is fixed as the loss function.

This model was used alongside of ensembles of decision trees and autoencoders for the selection of flat features.

5.2. Feature Selection Comparison

The following methods of feature selection were compared in an experimental study, the first three of which use the preliminary correlation optimization along the time axis for features:

- PLSC: hierarchical PLS clustering;
- Bayes: feature selection based on the Bayes subtrees built using the Chow–Liu method;
- LASSO: selection ordered by the magnitude of the coefficients of the constructed LASSO model;
- LightGBM: selection based on the importance of flat features in building an ensemble of gradient boosting trees (sliding window data representation);
- StemGNN: selection based on the attention matrix calculated inside a StemGNN network with a single block.

The following feature selection parameters were fixed for this and following experiments: PLSC with depth of splitting fixed as 4, Bayessian tree with depth of feature subtree fixed as 4; LASSO with top 16 selected variables by absolute coefficient value; StemGNN trained on spline interpolated data (order is 3) with top 16 selected variables by attention matrix; and LightGBM with features used more than once. The models were built on the whole filtered data set.

Below, Table 3 presents the variables selected by the proposed and existing methods, sorted by the number of methods that selected each variable. Only variables that have been selected by more than one method are presented. The variable marked by the field expert as important for the operation of the system as a whole is marked bold. It is worth noting that

Fig. 16. RNN model with StemGNN feature selection.

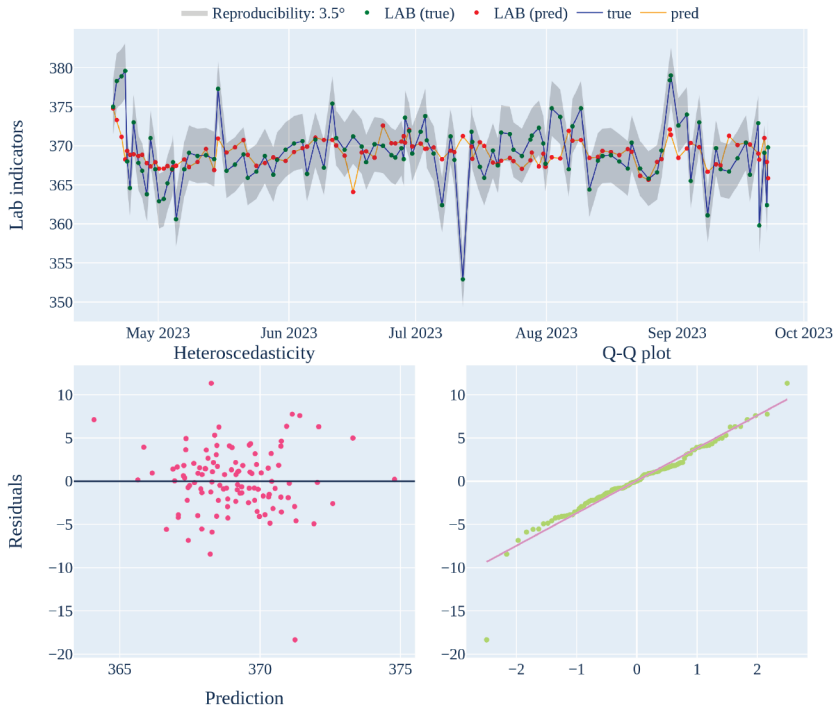

Fig. 17. RNN model with PLS feature selection.

a significant number of unrepresented variables were selected by only one method, which probably confirms the presence of a large number of correlated variables in the data set.

The Bayes feature selection method seem to have the most influence among the competitors, as the one to select all but one present sensor variables, while the LightGBM and StemGNN methods seem to make the most unique choices among the proposed approaches.

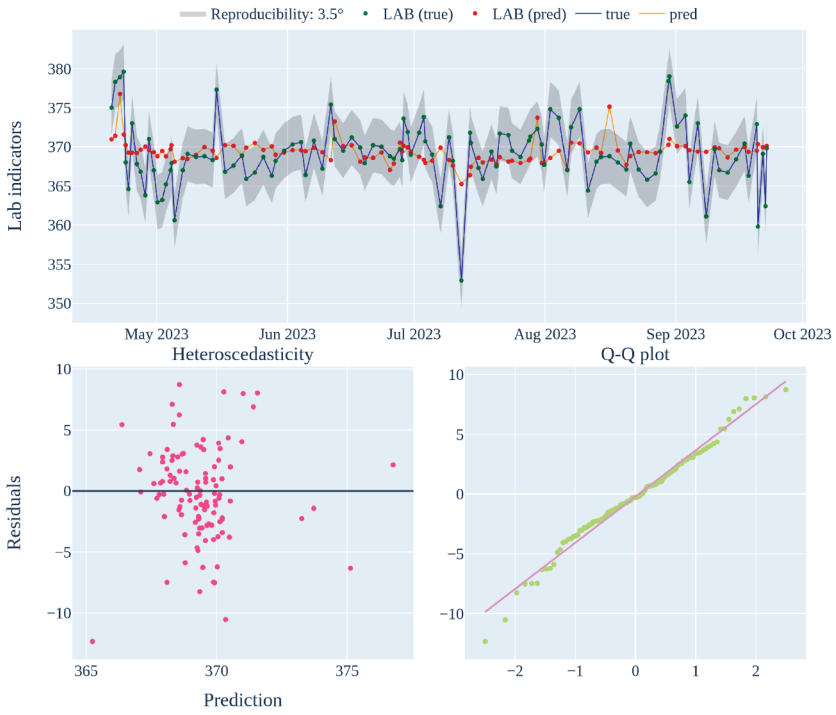

Fig. 18. RNN model with Bayesian feature selection.

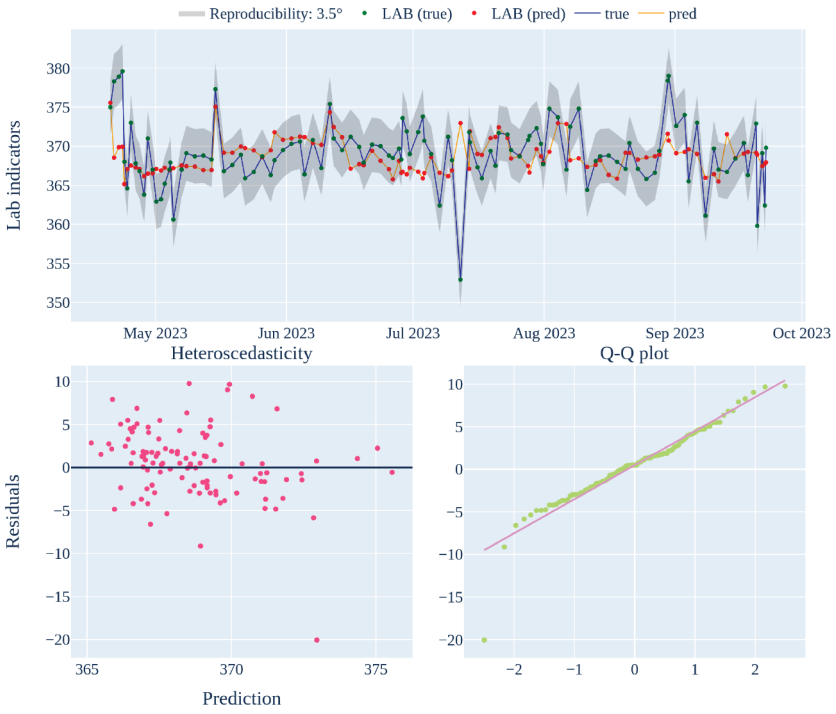

Fig. 19. RNN model with correlation-based L_2.

5.3. Quality Metrics

The problems of estimating the forecast of laboratory indicators quality include such factors as: the lack of real laboratory values to evaluate the approximation based on the constructed models, the presence of outliers in real data, inaccuracy of laboratory measurements—the presence of reproducibility limits. We also note that the critically small amount of input data, coupled with their temporary structuring,

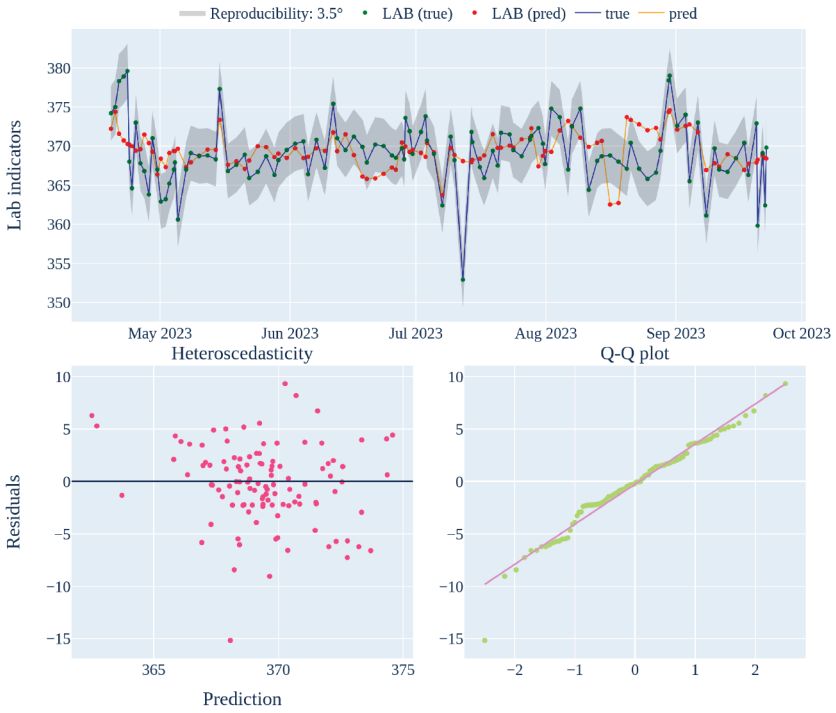

Fig. 20. Simple fully connected model based on an autoencoder.

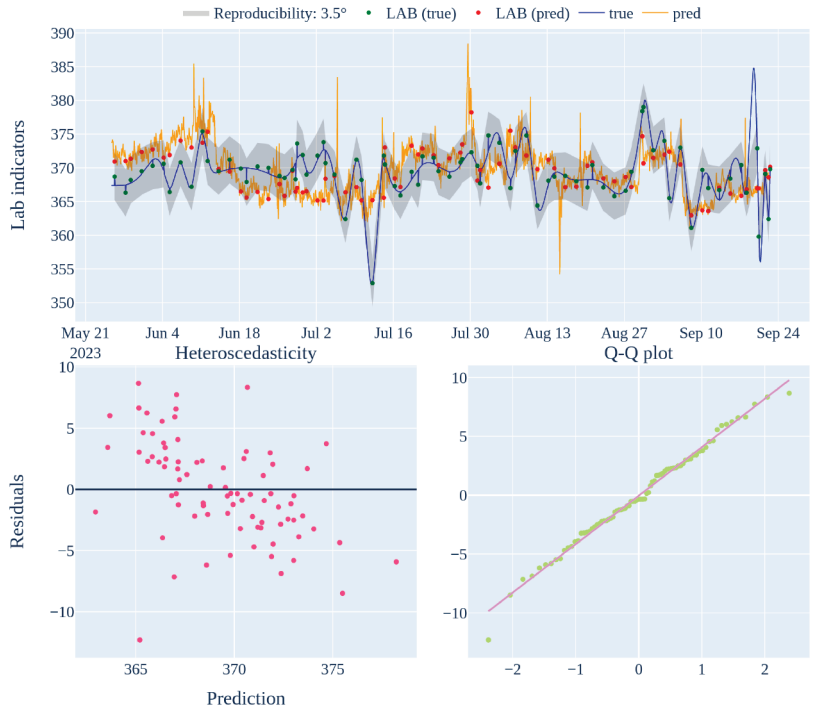

Fig. 21. StemGNN model trained on the spline interpolated data with PLS feature selection.

means that it is impossible to use such estimate instruments as cross-validation.

Hence, we offer a multistep approach to evaluating and comparing models. The first step is a rule of thumb, which we define by comparing the p-value for the hypothesis of the absence of linear correlation between the model results and real data with a fixed threshold. Alternatively, it is possible to investigate the adequacy by the R^2 threshold, however, for this data set, a correlation version is proposed. To take

Table 4. Experimental results for forecasting sorted by the values of the MSE metric

Feature	Model	Data	Pearson	Hinge	MAE	rMSE
StemGNN	RNN	Raw	**0.4055**	0.5901	**2.6995**	**3.7199**
PLS	RNN	Raw	0.3440	0.6894	2.8013	3.8207
AE	MLP	Raw	0.3823	0.7562	2.9069	3.8238
BAYES	RNN	Raw	0.3352	0.7542	2.9151	3.8326
	StemGNN	Loess	0.1379	**0.3213**	2.9092	4.0313
PLS	StemGNN	Spline	0.3209	0.9907	3.2494	4.0399
L_2 (Spearman)	RNN	Raw	0.2510	0.8128	3.0104	4.1021
LGBM	RNN	Raw	0.1946	0.8743	3.2158	4.1074
BAYES	StemGNN	Spline	0.3047	1.1974	3.5676	4.4891
LASSO	StemGNN	Spline	0.2665	1.3757	3.8468	4.8834
LASSO	RNN	Raw	0.2181	1.4014	3.8575	5.1372
LGBM	StemGNN	Gauss	0.2384	1.8203	4.8190	5.8204
LGBM	MLP	Raw	0.1566	2.8989	5.7509	7.3012

into account the inaccuracy of laboratory measurements (repeatability, see Subsection 3.2), it is proposed to use a modified Hinge metric for continuous response. Let the upper and lower limits of the measurement correspond to a symmetric interval of length I, then we can write down:

$$\text{hinge}(y, \hat{y}) = \max(|y - \hat{y}| - I, 0).$$

Alternatively, the models are compared using the basic L_1 or L_2 norms in the form of mean absolute and squared errors. The suggested comparison between models proceeds as following:

$$\text{Pvalue} \to \text{MSE} \to \text{Hinge} \to \text{Visual comparison}.$$

5.4. Laboratory Data Approximation Comparison

Let us describe an experiment comparing all the proposed and investigated methods of feature selection applied to the described above neural network models implementing both approximation and prediction. On the data preprocessed for stability (see Subsection 4.1), we select sequential training and test samples in the ratio of 6 : 4. Note that, due to the critically small amount of both test and training data, this is the only partition that we will limit ourselves to in this study.

Methods of feature selection with implicit time component use preliminary feature shift based on sorting lags using Spearman correlation p-value threshold: PLS clustering (PLSC), Bayes selection, Lasso, LightGBM. Methods with explicit time component such as sliding-window LightGBM, autoencoder, as well as automatic time component (StemGNN) are also being investigated. Feature selection methods use the same parameters as declared in the Subsection 5.2.

We fix the sliding window size of the forecast model inputs (as well as LightGBM model variation) as 24 h, fix the radius of the search for the optimal time component with the same value (the shift is applied only forward to prevent looking into future), as mentioned above, a time step forward of 1 h is investigated. Interpolation methods of laboratory data including Spline (order is 3), Gaussian kernel (with kernel size 24) and LOWESS (fraction of used elements is 2%) are investigated where needed, in each case the maximum offset between the real laboratory values is fixed as 48 h.

Table 4 shows the step-by-step results of the best models for each of the tested architectures used with proposed feature selection methods. As we can see, provided metrics mostly correlate excluding Hinge, resulting in the three groups of favourites—RNN model with StemGNN, PLSC, and Bayes feature selection, providing best correlation; multilayered perception model (fully connected) with autoencoder; and a Hinge metric leader, StemGNN model without feature selection, although it shows better correlations using proposed feature selection approaches (PLSC, Bayes).

The visual comparison for a select few of the described methods, as well as some residual visualizations can be seen in Figs. 16–21. Each plot has the

original lab measures plotted alongside the predictions and a reproducibility cone. The heteroscedasticity and Q–Q plots for predictions are presented to better demonstrate the errors.

6. CONCLUSIONS

This work has explored the problem of approximation and prediction of chemical laboratory indicators of the oil refining process, hindered by a small amount of data on the target variables. Laboratory indicators is the only way to acquire the quality of the manufactured product, however, they are not feasible in the form of physical sensors of the technological process. Therefore, the solution to this problem is sought in the form of a mathematical model (soft sensor) based on the physical sensors of the process, which come in large dimensions, since they characterize the technological process as a whole. The key problem of the study turned out to be the selection of features in the presence of a large number of correlated variables and a relatively small volume of the response records, complicated by an additional task of utilizing the expert knowledge.

The authors propose several new approaches that allow preprocessing the source scarce data for use inside complex neural network models, which, in the case of high dimensional data, have the bias of converging to the average value. We have proposed approaches based on correlation (in particular, partial least squares hierarchy), probability trees and graph neural networks, which were tested on popular neural architectures for working with time series within the framework of the studied data set. Compared to the existing studies, we focus on the time dependences between the physical and laboratory variables.

Separately, we highlight the proposed means of taking into account expert opinion for the potential use of the developed feature selection tools in real production. The developed methods might be implemented as a recommendation system for an expert (suitable for handing large amounts of process variables with multiple visualizations) as well as an independent automated tool. This should allow the ensemble of feature selection and tested neural network models to be used together with existing control systems to ensure smooth and effective transition to intelligent production plant management algorithms.

As a result, the authors also propose several best combinations of the proposed methods, which show the best results on the studied real data set and may take the expert opinion into account. Developed methods of feature selection based on PLS and Bayes approaches show best quality in combination with simple recurrent networks; using complex neural models for feature selection shows good quality as well. The main result of this work is a step in the development of the industry towards a recommendation expert system that allows approximating chemical production indicators in real time.

FUNDING

This work was supported by the Intellect Foundation.[2] The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

CONFLICT OF INTEREST

The authors of this work declare that they have no conflicts of interest.

REFERENCES

1. X. Zhu, Kh. U. Rehman, B. Wang, and M. Shahzad, Sensors **20**, 1771 (2020). https://doi.org/10.3390/s20061771
2. F. Curreri, S. Graziani, and M. G. Xibilia, Inf. Sci. (N.Y.) **537**, 1 (2020). https://doi.org/10.1016/j.ins.2020.05.028
3. A. Nair, A. Hykkerud, and H. Ratnaweera, Water **14**, 332 (2022). https://doi.org/10.3390/w14030332
4. Ch. Ou, H. Zhu, Yu. A. W. Shardt, L. Ye, X. Yuan, Ya. Wang, and Ch. Yang, IEEE Trans. Neural Networks Learn. Syst. (2022). https://doi.org/10.1109/tnnls.2022.3144162
5. R. Xie, N. M. Jan, K. Hao, L. Chen, and B. Huang, IEEE Trans. Ind. Inf. **16**, 2820 (2019). https://doi.org/10.1109/tii.2019.2951622
6. Y.-L. He, X.-Y. Li, J.-H. Ma, Sh. Lu, and Q.-X. Zhu, J. Process Control **113**, 18 (2022). https://doi.org/10.1016/j.jprocont.2022.03.008
7. J. Jiang and J. S. Rao, Annu. Rev. Stat. Its Appl. **7**, 337 (2020). https://doi.org/10.1146/annurev-statistics-031219-041212
8. A. E. Stott, S. Kanna, D. P. Mandic, and W. T. Pike, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, 2017* (IEEE, 2017), p. 4177. https://doi.org/10.1109/icassp.2017.7952943
9. Ch. Ding and X. He, in *ICML'04: Proceedings of the Twenty-First International Conference on Machine Learning, Banff, Canada, 2004* (Association for Computing Machinery, New York, 2004), p. 29. https://doi.org/10.1145/1015330.1015408
10. J. Tian, X. Sun, Yu. Du, Sh. Zhao, Q. Liu, K. Zhang, W. Yi, W. Huang, Ch. Wang, X. Wu, M.-H. Hsieh, T. Liu, W. Yang, and D. Tao, IEEE Trans. Pattern Anal. Mach. Intell. **45**, 12321 (2023). https://doi.org/10.1109/tpami.2023.3272029

[2] https://intellect-foundation.ru/.

11. D. Cao, Yu. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Yu. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, in *Advances in Neural Information Processing Systems*, Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Curran Associates, 2020), Vol. 33, p. 17766. https://proceedings.neurips.cc/paper_files/paper/2020/file/cdf6581cb7aca4b7e19ef136c6e601a5-Paper.pdf.
12. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, in *Advances in Neural Information Processing Systems*, Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, 2017), Vol. 30, p. 3149. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.
13. L. Freijeiro-González, M. Febrero-Bande, and W. González-Manteiga, Int. Stat. Rev. **90**, 118 (2022). https://doi.org/10.1111/insr.12469
14. F. Guo, R. Xie, and B. Huang, Chemom. Intell. Lab. Syst. **197**, 103922 (2020). https://doi.org/10.1016/j.chemolab.2019.103922
15. Sh. Yang, X. Yu, and Yi. Zhou, in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, 2020* (IEEE, 2020), p. 98. https://doi.org/10.1109/iwecai50956.2020.00027
16. D. Xu, Sh. Zhang, H. Zhang, and D. P. Mandic, Neural Networks **139**, 17 (2021). https://doi.org/10.1016/j.neunet.2021.02.011
17. E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, Multimedia Tools Appl. **82**, 16591 (2023). https://doi.org/10.1007/s11042-022-13820-0
18. V. S. Lalapura, J. Amudha, and H. S. Satheesh, ACM Comput. Surv. **54**, 91 (2021). https://doi.org/10.1145/3448974

AI tools may have been used in the translation or editing of this article.