



A framework for learning and embedding multi-sensor forecasting models into a decision support system: A case study of methane concentration in coal mines

Dominik Ślęzak^{a,*}, Marek Grzegorowski^a, Andrzej Janusz^a, Michał Kozielski^b,
Sinh Hoa Nguyen^c, Marek Sikora^{b,d}, Sebastian Stawicki^a, Łukasz Wróbel^{b,d}

^a Institute of Informatics, University of Warsaw, ul. Banacha 2, Warsaw 02-097, Poland

^b Institute of Informatics, Silesian University of Technology, ul. Akademicka 16, Gliwice 44-100, Poland

^c Polish-Japanese Academy of Information Technology, ul. Koszykowa 86, Warsaw 02-008, Poland

^d Institute of Innovative Technologies EMAG, ul. Leopolda 31, Katowice 40-189, Poland

ARTICLE INFO

Article history:

Received 31 May 2017

Revised 18 February 2018

Accepted 3 April 2018

Available online 4 April 2018

Keywords:

Sensor data processing

Methane concentration forecasting

Sliding-window feature engineering

Feature subset ensemble selection

ABSTRACT

We introduce a new approach for learning forecasting models over large multi-sensor data sets, including the steps of sliding-window-based feature extraction and rough-set-inspired feature subset ensemble selection. We show how to integrate this approach with the major data-processing-related components of DISESOR – a decision support system which is a coherent and complete framework for exploring streams of sensor readings registered in underground coal mines. As a case study, we report our experiments related to the task of methane concentration forecasting. The contributions in this paper refer to both the analysis how the nature of sensor readings influenced the architecture of the developed system and the empirical proof that the designed methods for data processing and analytics turned out to be efficient in practice.

© 2018 Elsevier Inc. All rights reserved.

1. Introduction

In the last decade, intensive growth in capabilities and popularity of analytical environments containing data mining solutions has been observed. Marketing, insurance, banking and finance, trade (especially e-commerce) and health care are the most popular applications. Less frequently, data mining methods are used to analyze and supervise industrial processes. The industrial monitoring systems usually produce multivariate streams of sensor readings for which performing standard preprocessing steps (such as data integration, data cleaning, feature extraction, etc.) is quite challenging. It is also difficult to construct and maintain forecasting models that should be used in an on-line fashion in industrial decision support systems. Nevertheless, potential benefits coming from intelligent utilization of this data source are truly huge.

In this paper, we investigate new approaches for learning forecasting models from multi-sensor data, for the purposes of monitoring natural hazards and industrial processes. We discuss them from the viewpoint of our decision support system – called DISESOR – which comprises of the expert system shell with the knowledge base that can be used together with the data incoming on-line, the feature engineering module that can derive the most meaningful statistics describing multivariate

* Corresponding author.

E-mail address: slęzak@mimuw.edu.pl (D. Ślęzak).

sliding time windows, as well as the prediction module that enables the users to easily define, monitor and adapt specialized model parameters based on thorough analysis of the collected data.

The architecture of the considered system is designed to fit various fields. However, the area of our interest is the mining industry (coal, salt, copper mining, etc.), with an emphasis on underground coal mining. Our motivation for considering intelligent forecasting models in this particular area of applications raises from the fact that coal mines are well-equipped with monitoring, supervising and dispatching systems connected with machines, devices and transport facilities. There are also some systems aimed at monitoring natural hazards (methane, seismic and fire hazards). However, the collected information is typically used only for raw data visualization purposes, whereas a deeper data analysis could significantly improve many coal mining processes.

The developed system consists of several architectural layers. From a data integration perspective, in its heart there is a relational data repository where all detailed measurements collected from different sources are stored. It was actually required to design a common interface allowing to describe in a consistent way a variety of types of sensors and devices. It is important to stress that this kind of unification – referred here as the *sensor card* – was not done before in the area of underground coal mining. On the other hand, all functionalities of our system depend on it heavily. In particular, the new ensemble-based feature selection and forecasting model learning methods described in this paper would have no chance to work efficiently without such prior step of data integration.

On top of the data repository, our system includes the data preparation and cleaning layer, as well as further modules responsible for data analytics, prediction and support. Our interest is in the process of transforming and utilizing the original sensor readings to forecast near-future status of monitored coal mine areas. One can think about such forecasts as *virtual* (or *soft*) sensor readings, i.e., sequences of values that are predictions of future measurements. High-quality predictions may be then used as inputs to the expert system that advises and warns the humans. The efficiency of predictions depends on the quality of repository contents, as well as the robustness of data processing and machine learning techniques embedded into the system. It is expected that particular forecasting models are trained off-line on the repository data and then applied on-line against multivariate streams of the incoming readings. It is also expected that the system users can influence the processes of both off-line learning and on-line deployment.

We focus especially on the feature extraction and feature selection methodologies which address the problems of, respectively, deriving useful and understandable parameters (variables, attributes) from raw sensor readings and reducing the amount of those parameters in order to achieve possibly simplest yet accurate models. With regard to feature extraction (or feature engineering), we attempt to describe the data in a possibly intuitive way, using statistics derived from sliding time windows. In the case of underground coal mine sensors, derivation of multivariate series of simple window-based statistics allows us to deal with noisy and incomplete data sources, better reflect temporal drifts and correlations and reliably describe real situations using higher-level data characteristics. Thus, the off-line and on-line sliding window operations are an integral part of data preparation and cleaning.

The task of feature selection may be defined at two levels. First, in the case of predicting near-future readings of a particular sensor, one could think about it in terms of choosing other relevant sensors and time frames that are sufficient to start the process of model training. In this case, it is crucial to interact with domain experts and provide appropriate analytic reports/visualizations to let them make the right assessments. This level might be also called an information source selection. The second level refers to selecting specific features constructed during the time windowing process. Herein, as proved in this paper for the case study of medium-term methane forecasting in the mining excavation, one can successfully proceed with very simple rough-set-based methods which yield surprisingly small feature sets. Definitions of selected features can be then registered as parameters within the data preparation and cleaning module in order to establish an efficient framework for deploying, monitoring and tuning the forecasting models embedded into the system.

In summary, there are three major contribution aspects of our work. The first one is a novel decision support system framework which adapts and expands some of the state-of-the-art analytical tools. According to our knowledge, this is the first architecturally complete and easily extendable system in this area. The second aspect is the above-discussed sensor card – an information template created on the basis of investigation of a large variety of sensors that can be applied whenever heterogeneous data sources need to be systematized. The third contribution relates to a new approach to solving the above-mentioned problem of methane forecasting and, more generally, managing prediction-oriented virtual sensors. The proposed solution utilizes the readings of multiple sensors and is focused on intuitive feature extraction/selection methods. Its advantage is that it operates on just a few regression models defined on diversified feature subsets. Unlike typical ensemble-based machine learning techniques, this particular approach is truly simple in maintenance and provides the system users with good means for interaction. On the other hand, the experimental results obtained on real-life data sets show that its prediction efficiency is not worse than in the case of random forests, support vector machines, or deep learning models.

The paper is organized as follows. [Section 2](#) presents an overview of previous research related to the presented topics. [Section 3](#) outlines a general architecture of the introduced system. [Section 4](#) focuses on the above-mentioned sensor card specification and organization of collected sensor readings in a relational data warehouse. [Section 5](#) introduces our approach for time-window-based data preprocessing and feature extraction. [Section 6](#) presents the basics of our prediction module and some examples of using its outcomes. [Section 7](#) presents the case study of methane concentration forecasting, including the analysis of the results of an open international data mining competition that we organized to validate correctness of architectural decisions while designing the data processing components of our system. [Section 8](#) outlines the feature

selection algorithms that we implemented for the purpose of constructing forecasting models that would be equally accurate but simpler than those designed by the competition participants. Section 9 reports an experimental comparison of our own methods versus the competition winners, conducted both on previously published and unpublished real-life data sets gathered in the currently functioning underground coal mines. Section 10 concludes the paper.

2. Related work

This section includes some examples of state-of-the-art solutions in the areas relevant to our contributions in this paper. We discuss relationships of our research to the fields of: decision support, industrial data processing, monitoring natural hazards, time series analysis, window-based data aggregations, feature extraction and selection, as well as construction of forecasting models.

Decision support systems are applied in many fields related to optimization of various logistics and planning processes, monitoring and risk management, etc. [32]. One can also find some elements of decision support solutions dedicated specifically to the mining industry [41].

From our perspective it is particularly important to study decision support applications in non-stationary environments. Let us refer to the works on concept drift in business intelligence and data-driven decision support systems, with an example of a data-distribution-based concept drift detection method proposed for data streams in [11]. As another example, let us consider an adaptable decision support system integrated with a wireless sensor network developed for the purpose of autonomous closed-loop zone-specific irrigation, equipped with a rule induction mechanism that extracts new knowledge related to a plant state diagnosis [17]. It is also worth referring to real-time decision support systems that integrate the adaptive intelligent forecasting models with the knowledge base and decision making layers in an information agent fashion [43]. Finally, let us mention a decision support system with multiple adaptive prediction models that take into account the appearance of new (or deletion of old) sensors developed successfully in the chemical industry [5].

Decision support systems in industrial applications are closely related to explanatory and predictive analysis. Industry processes are typically monitored by a number of sensor devices. This automated monitoring allows for appropriate performance control of a process and facilitates further data analysis involving various aspects of modeling and prediction of sensor readings [14]. In the same time, it is important to design appropriate data management solutions in order to gather both original sensor readings and prediction results, taking into account general decision support system requirements, as well as semantics of data processed in particular applications [1]. In the proposed architecture, we attempt to follow the state-of-the-art approaches in both of the above areas in order to maximize benefits coming out of modern data analytics.

Applications of data mining methods in the area of monitoring industrial processes are challenging because of the nature of sensor readings [3]. It is therefore important to rely on well-examined approaches in the area of data quality assurance, including data integration and aggregation. A general overview of multi-sensor data fusion methods can be found in [29]. Some aspects of dealing with inconsistent, sensor readings were addressed in [37]. For some examples of challenges present in the specific area of underground coal mining, let us refer to [52]. In our system, we attempt to follow the experience of other researchers. However, comparing to other approaches, we pay an additional attention to data modeling and careful categorization of new data sources.

Up to now, there are just a few research groups active in the field of intelligent monitoring of natural hazards in coal mines [33]. In this paper, we consider the task of forecasting methane concentrations from the viewpoint of both accuracy and further utilization of obtained predictions. For another example of forecasting tasks in coal mining, we refer to seismic hazard prediction methods reported in [28]. Yet another example of using machine learning and artificial intelligence techniques in mining-related forecasting can be found in [27]. Other aspects related to monitoring and managing natural hazards in the mining industry are discussed in [35]. All such studies are an important inspiration while developing decision support systems. In particular, our system is designed so as to easily embed both the existing and new forecasting models.

While investigating sensor readings, one can rely on widely-known methods of multivariate time series analysis, including window-based segmentation and aggregation [15]. For an example how to integrate multi-sensor time series analysis with external sources of spatio-temporal information, let us refer to [46]. An example of utilization of such statistics as higher-level features in time-series-based data mining can be found in [51]. An interesting industry-specific case study of deriving wavelet-based features from sliding time windows is also reported in [12]. When compared to the other decision support systems, our approach is the first successful attempt in the area of coal mining, whereby the forecasting models rely fully on such types of sliding-window-based features.

Deriving statistics from sliding time windows can be regarded as an example of feature engineering which is one of crucial steps in any knowledge discovery process [10]. In our studies, we pay equal attention to the stages of such understood feature engineering and feature selection which is responsible for extracting a relatively small subset of (engineered) features that are both the most relevant and sufficient to solve the investigated problem [9]. An example of application of feature selection in the area of sensor data analytics can be found in [8]. From the point of view of our research, the most useful methods are those based on so-called filter strategy because of its computational efficiency and independence from the particular prediction algorithms. In particular, let us refer to the Minimum Redundancy Maximum Relevance method (mRMR) [40] which inspired us to develop some of functionalities available in our system.

Another source of inspiration came from the theory of rough sets, in relation to various formulations of a decision reduct [38]. Decision reducts are irreducible subsets of features that provide (almost) the same level of information about de-

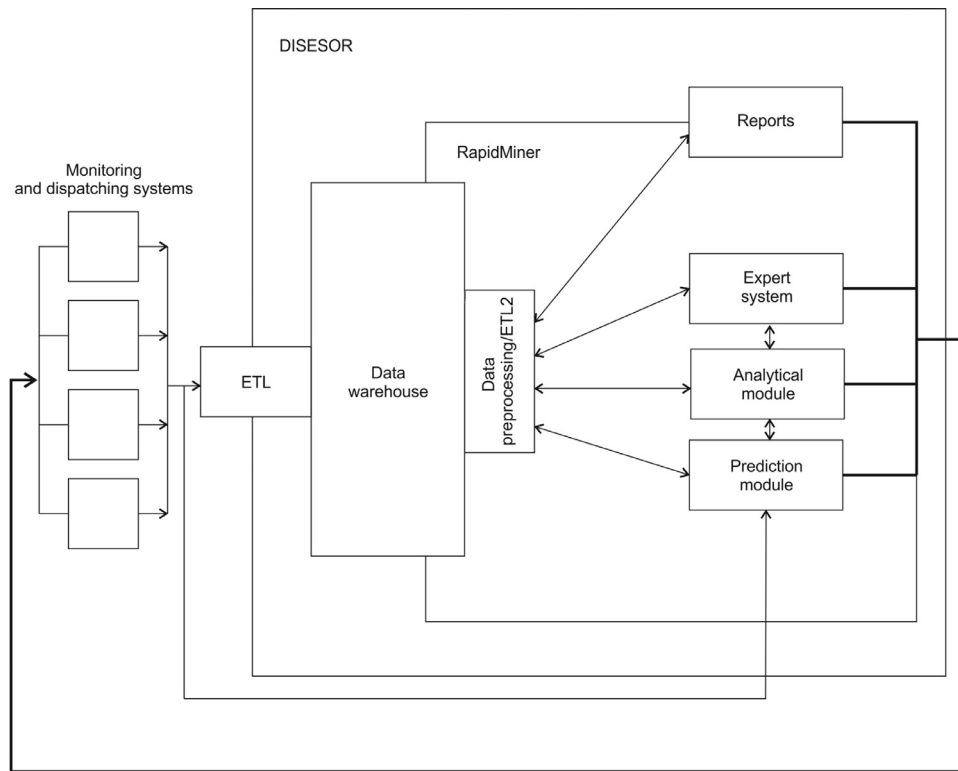


Fig. 1. The architecture of the considered decision support system.

pendent variables as the complete feature set. The level of information is often expressed by means of data records with different values of a dependent variable that become discerned by particular feature subsets. Among extensions of standard rough-set-based feature selection approaches, let us mention about discernibility-based feature clustering and algorithms aimed at searching for larger collections of diversified reducts [20]. Given the dynamics of the data that we are dealing with, it is also worth referring to some approaches to rough-set-based incremental feature selection [26]. From the viewpoints of working with feature clusters and incremental calculations, one can also compare rough set methods with some of the mRMR extensions [34].

As some of readings can be temporarily unreliable, it is also worth studying extensions of standard feature selection algorithms aiming at constructing more robust models [2]. Moreover, one can work with feature subset ensembles which are somewhat analogous to the above-mentioned collections of reducts [42]. From the perspective of the evolving data, it is also important to combine the task of feature selection/extraction with the state-of-the-art approaches addressing the aforementioned concept drift [16]. It may be particularly useful to search for features that are indifferent with regard to the drifting data and treat them as a starting point for constructing robust forecasting models [7]. This last-mentioned strategy was utilized in our system as well.

3. Conceptual architecture

Our system can process sensor readings that are collected by multiple monitoring and dispatching modules deployed in different coal mines. Its architecture is fully modular and, therefore, we believe that its high-level design could be reused also outside the field of coal mining. It actually follows a general industry trend of designing data-driven decision support systems [1]. From the information flow perspective, its overall structure is presented in Fig. 1. It consists of the following major components:

- Data warehouse (DWH): A repository where sensor readings from different monitoring and dispatching systems are integrated and stored.
- Data preparation and cleaning module (ETL2):¹ A layer of processes aiming at preparing the data for further analysis.
- Modules that can utilize data coming from both DWH and directly from coal mine systems: Analytical, prediction and expert system modules.

¹ One might think about this module as a typical *Extract Load and Transform* layer (ELT). However, we regard abbreviation *ETL2* as easier to remember, as this is the second data preprocessing phase conducted after standard ETL routines.

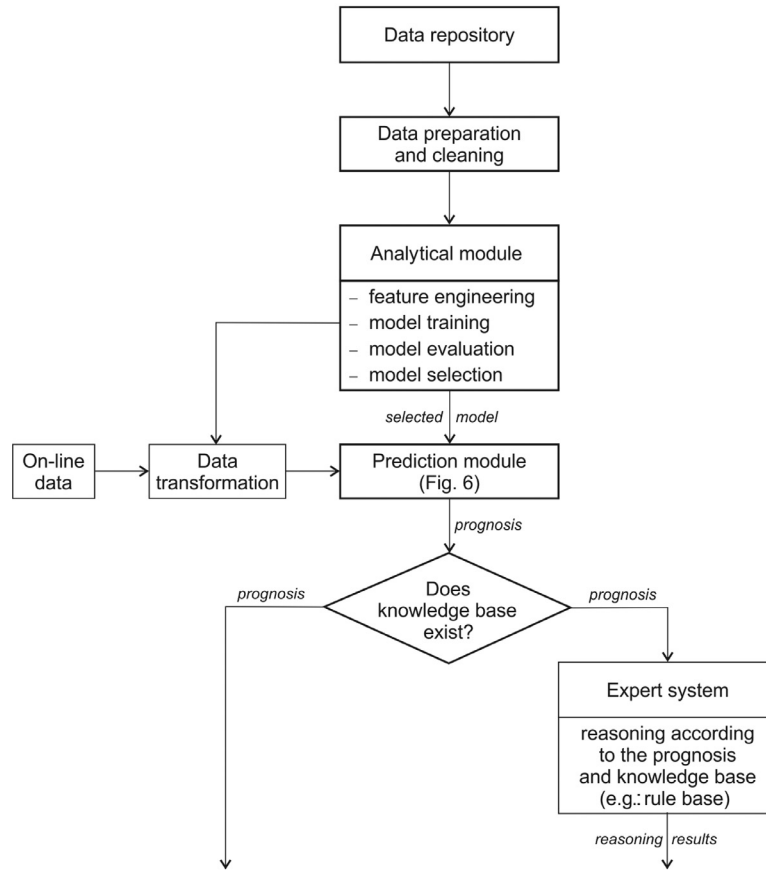


Fig. 2. Creation and deployment of a forecasting model.

The system components responsible for data preparation, cleaning, analytics and prediction were implemented within the RapidMiner environment [21], by means of adapting its standard components or developing new compatible methods. As a result, all the above phases are performed as the RapidMiner *operators* that meet the requirements of the coal mining decision support system users. Such operators can be applied multiple times in an unrestricted order. It is also possible to preprocess the data by means of basic RapidMiner operators that are dedicated to multivariate analysis, identification of outliers and missing values, etc. Moreover, we extended RapidMiner by additional operators wrapping some useful libraries written in R [31].

We will go back to the ETL and DWH settings in Section 4. For now, let us take a closer look at the ETL2 module. From a data flow perspective, its aim is to deliver integrated data sets based on the chosen sources in the selected time range. It also includes the tools responsible for data aggregation and unification, optional missing values imputation, as well as calculation of values of time-window-based features defined as a result of running the feature extraction and selection algorithms [15]. The outcomes of ETL2 are transmitted to the prediction module which can enrich them with additional features corresponding to the expected future sensor readings (referred as virtual sensors, see Section 1). Such data sets can be then utilized by diagnostic processes within the expert system module or transferred to the analytical module for further investigation. Some examples of algorithms currently implemented as a part of our analytical module's functionality can be found in Section 8.

In order to assure smooth ingestion of subsequently incoming data, all the above ETL2 settings are saved in compliance to the XML-based RapidMiner standard. Therefore, all components of our system are able to transform new data portions (collected on-line from the monitoring systems), so their final form is acceptable by our forecasting and inference solutions. From a deployment perspective, the overall framework is easy to maintain and modify whenever one develops any new forecasting models that are supposed to perform better than those currently embedded into the prediction module.

Fig. 2 presents a typical creation and deployment scheme of a forecasting model in DISESOR. After loading the data into the system repository, different models can be tested. It enables the users to verify application of various operators dedicated to our system (such as RuleInduction [48] and RapidRoughSets [25]), as well as all other operators provided by the RapidMiner environment. Next, the selected model can be loaded to the prediction module and applied to analyze the

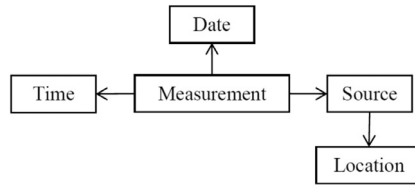


Fig. 3. DWH structure – simplified table schema.

Table 1

DWH structure – contents of data tables.

Table	Description
Measurement	Value of measurement
State	State of measurement (e.g.: alarm, calibration, breakdown)
Discretization	Measured values can be additionally discretized
Time	Time of measurement, range [00:00:00, 23:59:59], 1 s resolution
Time_category	Category (e.g.: time of mining)
Date	Date of measurement
Location	Location of measurement source
Location_attribute	Location characteristics
Location_hierarchy	Location hierarchical structure
Source	Measurement source (sensor or device)
Source_attribute	Source characteristics

data incoming directly to this module. We transform the raw data to a form expected by a given model. The resulting predictions can be treated either as a final result or as an input to the expert system's knowledge base.

4. Integration and storage of sensor readings

Our system integrates data from many other systems responsible for seismic or seismoacoustic monitoring, machinery monitoring, control of underground atmosphere, etc. Having all those data sources in a single place, provides a valuable opportunity to improve human safety and increases work efficiency [29,37]. However, those systems are provided by many different corporations and are developed using various technologies that are often focused on different types of sensors and different strategies of their utilization.

We started the overall design process with a review of the existing measuring devices. It included over 40 different types of sensors such as methane meters, anemometers, carbon monoxide sensors and a variety of machine diagnosis sensors (conveyors, shearers, etc.). The analysis of common and distinguishing features of all those instances led us towards specification of the aforementioned sensor card – a data source registration protocol that allows to consistently describe most of devices used in the domain of coal mining.

As outlined in Section 1, such unification was done for the first time in this particular area of sensor applications. The sensor card includes three specification categories: basic information, metrological information and system information. Information provided by each of the items can be textual (e.g.: device name) or categorical (measured value: concentration, temperature, etc.; location: excavation, longwall, etc.). This kind of classification becomes useful at the stage of combining information delivered by different monitoring systems. Detailed description of the sensor card items (with examples of their contents in a particular case of methane meter) can be found in Appendix A.

The analysis of different sensor cards, as well as investigation of databases present in some of the existing monitoring systems, influenced the final structure of our data warehouse (abbreviated as DWH, see Section 3). It is modeled as a snowflake illustrated in a general form by Fig. 3. The full list of tables with their description is presented in Table 1.

The *Measurement* table stores all detailed sensor readings. Dimensions related to the *Measurement* table are *Date*, *Time* and *Source*. *Date* and *Time* describe when a given sensor reading was registered. Besides these components several additional aspects are included:

- *Date* additional fields: Day of week (number, name, abbreviation), week number, month (number, name, abbreviation), quarter, half-year.
- *Time* additional fields: Half of hour, half of minute, shift numbers (assuming that they start at 4 or 5 a.m.), quarter.

The *Source* table contains such information about sensors/devices as: name, description, type name, measured quantity, measurement unit, name of a system that transmits the data, range of measurements. Fields in the *Source* table correspond to the structure of the sensor card outlined in Appendix A.

The *Source* table is described by means of *Location* – the dimension table that describes where in a coal mine the measurement source is located. The *Location* table has a hierarchical structure. The top-most level of hierarchy is formed by the

coal mine divisions. Divisions consist of the seams which are divided into the mining areas. At the bottom of hierarchy there are the mining workings.

All tables are populated with new data records using the ETL processes comparable to those designed for the coal mine monitoring and dispatching systems deployed in Poland, Ukraine and China. The ETL routines were designed using Open Talend Studio [6]. During the performance tests, our DWH was loaded with 800 million records what resulted in 200 GB of data. It enabled the performance tests and optimization of both the logical data warehouse structure and the database management system (PostgreSQL). The *Measurement* table was partitioned according to the months of sensor readings and the indices for foreign keys in this table were created. The most frequently used SQL statements were tuned and thoroughly tested using the JMeter environment [13]. In particular, the time windowing and aggregation operations described in Section 5 are implemented as a combination of SQL and RapidMiner routines.

5. Data preprocessing and feature extraction

In this section, we outline our approach for feature extraction, aimed at processing the data obtained from sensors that monitor changes in the environment and provide outputs in a form of time series. Individual readings may take different forms according to the application domain [14,37]. Values may express continuous phenomena, such as pressure, humidity, or the level of methane concentration in a longwall of a coal mine. They can also express a discrete state of the environment, such as an on/off state of a device or vehicle movement direction. To acquire knowledge about the environment states, it is common to set up a collection of sensors, potentially of different types. Therefore, the gathered data elements can be complex on various levels and sometimes their interpretation is possible only in a context of additional domain knowledge, stored by means of the already-discussed sensor card.

Prior to feature extraction, let us outline some basic steps of data preprocessing which becomes specially important when real-life streams of sensor readings are involved. Measurements recorded by sensor devices tend to be noisy. Thus, the goal of data preparation and cleaning module (referred as ETL2, see Section 3) is to translate the data stored in DWH to a form acceptable by the forecasting model construction methods. In other words, the ETL2 module prepares the training sets for further analysis and, once the models are ready, becomes responsible for feeding them with new inputs. In particular, the ETL2 layer is designed to handle the following issues:

1. *Unsynchronized readings*: Reading frequencies differ for different sensors.
2. *Missed readings*: Sensors may stop delivering in a given time interval.
3. *Outlier readings*: Sensor readings are frequently imprecise or unreliable.
4. *Rare readings*: Usually, critical events occur in the data very sparsely.

The first task is to adjust sensor readings that are collected at different frequencies. Also, some systems collect a new reading only in the case of a sufficiently significant change of the measured value. While processing the data through the ETL2 layers, we need to proceed with all such aspects according to specifications written down in the sensor card (see Appendix A).

The second task is to impute the missing values and identify outliers. This is a particularly complex task for spatio-temporal data environments [4,46]. With this respect, one can utilize standard RapidMiner framework. For instance, one can create a logical expression that defines value replacements (e.g.: replace values < 1 with “low state”), use a default value, follow the last valid reading, take an average of the neighboring readings, apply linear regression based on the preceding values, etc. The maximum allowed number of consecutive missing values that can be imputed should be set up too. On the other hand, as we discuss further, the missing value imputation (or outlier replacement) is not always a requirement. It may depend on a context of a given sensor and knowledge about its operation conditions or, simply, on further preprocessing steps which can deal with incomplete data on the fly.

A special case of data preprocessing corresponds to the creation of a dependent variable. This is the crucial aspect for the supervised learning approaches. For this purpose, we implemented a dedicated operator which allows us to define a dependent variable as the maximal value measured for a given sensor within a specified time interval (e.g.: three to six minutes into the future, as in our case study reported in Section 7). Although such functionality seems to be basic, let us note that it was not present in the original RapidMiner environment. Actually, it can be considered as an example of a broader window-based methodology described below. Such a style of specifying a dependent variable may also decrease sparsity of its critical values. This is because a single high measurement influences a score of the whole time interval.

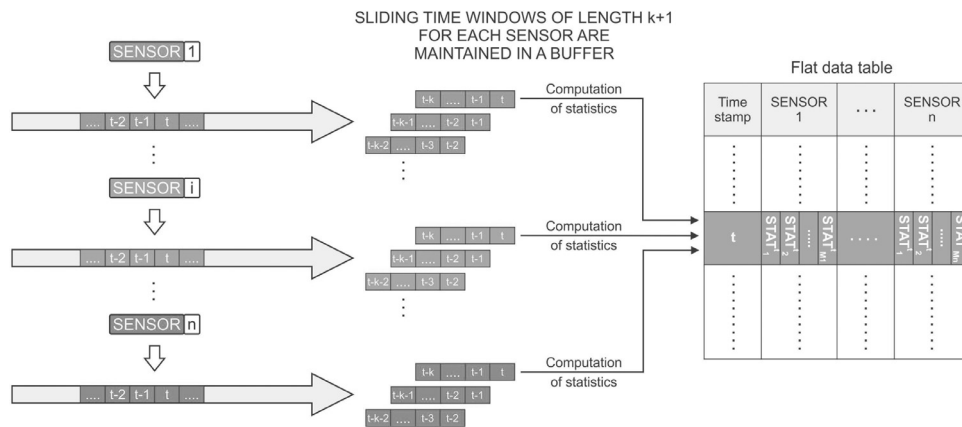
Given the above steps of data preprocessing, we are now ready to go back to the topic of feature extraction. In the proposed approach, a sliding time window method is utilized to overcome some of the above-mentioned issues related to the time series data. Window-based aggregation of sensor readings replaces a sequence of values with some of its statistics. The range of aggregation can be chosen by the users by means of, e.g., a time unit that defines windows containing sensor readings to be grouped together. For each aggregation outcome, we can calculate a weight that is inversely proportional to a number of involved missing values. This approach enables us to reduce the number of missing values in the data and introduce weights that can be utilized by further analytical methods. It is also worth mentioning that such aggregation operations can work on multiple sensor readings with unsynchronized frequencies.

A collection of sensors that perform readings produces a corresponding collection of the time series statistics. A single time series is an ordered sequence of readings associated with the timestamp at which it was collected. A collection of

Table 2

Examples of features that represent each time window.

Feature type	Description
Basic summary of all readings in a given time window	Simple statistics: mean, median, min, max, stdDev, 10th and 90th percentiles
Summaries of consecutive sub-windows	The same statistics as above, computed for sub-windows of a given window
Trends related to recent readings in a time window	Differences between the last reading and min/max values, differences between last/first readings in a window
Statistics for differentiated values in a time window	Mean, median, min, max, stdDev, 10th and 90th percentiles of differences between two consecutive readings
Measures derived from summaries of a time window	Differences between min and max, mean and median, max and 90th percentile, min and 10th percentile
Measures derived from summaries of sub-windows	Differences between quantities of mean, median, min and max representing consecutive sub-windows
Indicators of extreme readings	Position in a time window of a reading with min/max value, position of a min/max value in the latest sub-window
Indicators of extreme transformed readings	Position of a maximum difference between consecutive readings, etc.

**Fig. 4.** Window-based features calculated for a portion of the time series data.

aggregated values created from a time series may be organized arbitrarily at a higher level. For instance, if a time window covers one minute, then we may be interested in five consecutive windows that cover five minutes of data in order to analyze trends during that period. It is different from aggregations over a single five-minute time window. Going further, different sensors correspond to different time series threads processed independently. To obtain better description of the environment at a given time point, we need to combine sequences coming from different sensors and form a larger set of aggregations [19,22].

In the ETL2 module, processing of a single time series requires two parameters: *length* and *offset*. The first of them defines the size of a time window, e.g., the number of readings or a corresponding time interval. The alignment of processed time windows is controlled by the second parameter. It defines a degree in which any two consecutive windows overlap with each other.

During the process of moving a time window through a time series, each of its fixed positions defines so-called basic window. For such atomic slice of data, a predefined set of aggregation functions is applied. Each aggregation function can be seen as a new window feature. Default features that are calculated to represent the data in a basic window are presented in Table 2. If we consider more than one consecutive basic window to represent the environment state at a given time point, then we can extract some additional features expressing trends and changes between window pairs. Some examples of inter-window features that are calculated to represent the changes between two consecutive basic windows are presented in Table 2 too. Values of such features are assigned to a single basic window compared to its direct predecessor. Obviously, features derived from sliding time windows can vary depending on the area of application [12,50]. Nevertheless, the overall mechanism of computing time-window-based aggregate representations can be treated as a universal approach.

The proposed framework is capable of operating on multivariate time series derived from a number of sensors. The default method of processing multiple series is hierarchical, i.e., each time series is processed independently and then the results are combined according to specified settings. Afterwards, depending on configuration, features corresponding to basic windows and inter-window features derived for selected sensors create so-called composite windows which represent the overall state for a given time point. This way we equip our system with a comprehensive data preprocessing and feature extraction framework that can be used for constructing informative and robust representation of multivariate time series data, as visible in Fig. 4. Due to a diversity of extracted features and a high number of considered sensors this representation of data may be highly dimensional. It requires specific feature selection and forecasting model construction techniques to be

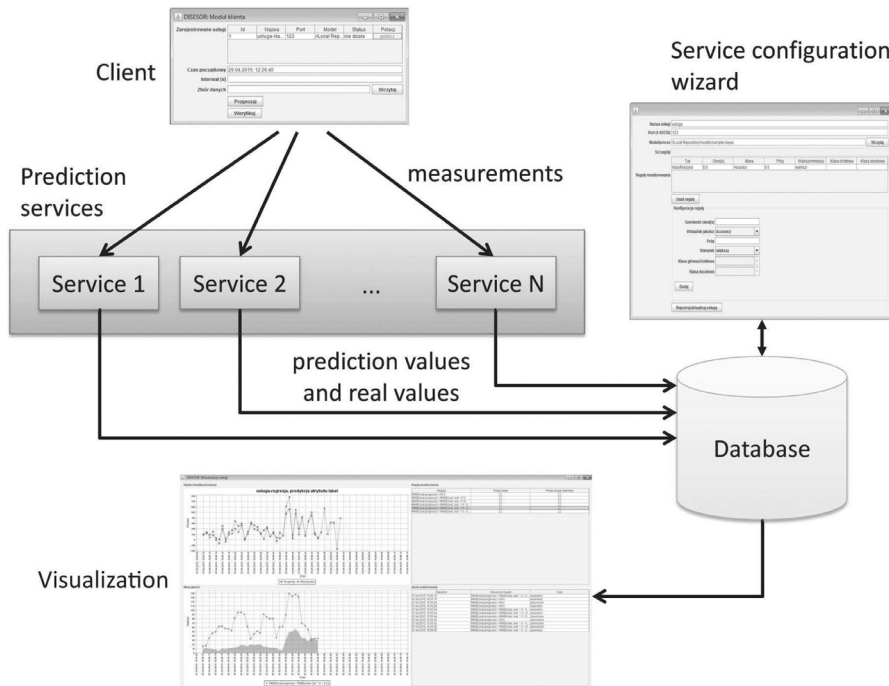


Fig. 5. Architecture and operational scheme of the prediction module.

applied next. For example, for the methane-related data set investigated in further sections, the total number of extracted features easily exceeds the amount of two thousands.

6. Prediction module and using its outcomes

Our prediction module is aimed at applying forecasting models trained over the ETL2-preprocessed data, for a given time horizon and a given frequency of considered sensor readings. The module provides interfaces that allow the users to select quality measures and their thresholds that ensure the minimal required prediction quality. If the thresholds are met, then predictions are treated as future values of a given sensor. The prediction module can also conduct an on-line batch re-learning of forecasting models and track quality measure trends in the incoming sensor readings [3,16].

As mentioned in Section 3, the prediction module is just one of examples of architectural layers that can be fed by the ETL2 outcomes. However, the quality of predictions is highly important for the whole system. Predictions can be visualized directly as virtual sensors or they can be used, e.g., as inputs to our expert system module that is aimed at performing on-line and off-line diagnosis of machines and other technical equipment. This module supports also decision making with respect to both technical condition of equipment and inappropriate execution of coal mining processes. However, all the implemented reasoning mechanisms work correctly only for valid inputs, including the outcomes of the prediction module.

From the architecture perspective, the prediction module is based on web services that are responsible for forecasting dependent variable values (discrete labels, continuous scores, etc.) on the basis of input vectors. Such services are connected with appropriate regression or classification models [22,48]. A typical scenario of a prediction service application is as follows:

1. Client sends a prediction execution request accompanied by a vector of selected window-based feature values and a timestamp.
2. Service loads calculated prediction to an internal RapidMiner database.

The database stores descriptions of forecasting models and required transformations of original sensor readings, as well as information about the training data sets used for constructing particular models, their parameters and previously-mentioned quality thresholds. Thanks to connecting it with the above services, it also stores the predicted values of dependent variables and it can compare them with actual values that become available after some time. Any instance of model modification/adaptation is stored as a new database entry which makes full history of changes available to the users.

Predictions can be visually compared to the actual values on a single plot (see Fig. 5). This visualization can be performed by a monitoring or dispatching system, where predicted values are delivered as readings of a virtual sensor. Visualizations of the system performance are essential to strengthen the user interactions in any type of a risk monitoring/management framework.

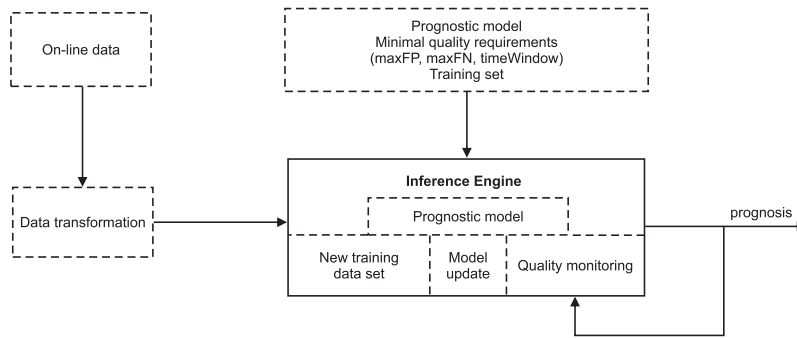


Fig. 6. Principles of the prediction module operation.

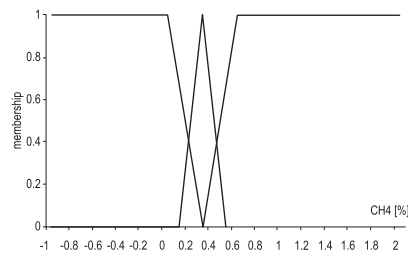


Fig. 7. Fuzzy set representation of the states of methane concentration.

The prediction module relies on a forecasting model, the training data set based on which the model was created and the values of quality thresholds that the model should meet. Prediction quality monitoring is based on a sliding time window within which the quality is verified (e.g.: one hour), frequency of prediction calculation (e.g.: one minute) and indicators that are typically called *FalsePositive* and *FalseNegative* (abbreviated as FP and FN). Their acceptable thresholds (abbreviated as maxFP and maxFN) are explicitly defined by the users for each of dependent variable values. Given a number of predictions and the FP/FN scores over a given time window, it is possible to calculate various standard quality measures of a forecasting model, such as the overall classification accuracy, g-mean, specificity, sensitivity, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), etc. [31,47]. For regression tasks, we support ε -insensitivity which means that the predictions that differ by less than a given threshold (ε) from the real values are not regarded as an error.

It is assumed that if the quality of predictions issued by a model decreases below a given threshold, then a new training set is automatically collected. The size of such new data set is the same as the size of the original training data. In this way, our system attempts to adapt available models to changing conditions in a mine. The adaptation is performed by modifying only the input features and the parameters of the preselected model. The learning approach itself is not changed. This means that once we chose, e.g., a regression approach for a given task, then – in the current implementation of our system – we cannot switch it automatically to a different method. Next, the quality of a newly trained model is verified on the data that triggered the adaptation mechanism. These cases are not included in the new training set. If the quality of the adapted model is satisfactory, then the new model is applied to predict future values of a dependent variable. Otherwise, a message is generated stating that the prediction cannot be continued and the system requires supervision of a data scientist. The overall process can be seen in Fig. 6.

Besides delivering the forecasts to a dispatching system, the predictions can be used as inputs to the knowledge base with fuzzy rules aimed at justifying the corresponding threats. For our case study of methane concentration, fuzzy reasoning can be conducted on the basis of its predicted levels and the dynamics of their changes. It is actually aligned with a more general trend of utilizing the elements of fuzzy logic in the risk monitoring and management systems [41]. In DISESOR, membership functions of fuzzy rules were worked out on the basis of interviews with the experts (dispatchers of various methane monitoring systems) and the corresponding national regulations on hazard estimation [18].

The concept of methane concentration was divided into four fuzzy sets corresponding to the following states: “normal”, “admissible”, “boundary” and “exceeded”. Depending on a kind of sensor and its location, its status can be related to different values of methane concentration. Fig. 7 illustrates fuzzy sets reflecting the states of methane concentration for sensors with the safety limit at 2% (see Appendix A). The dynamics of changes is represented by three fuzzy sets: “no changes”, “increasing” and “quickly increasing” (Fig. 8). Finally, there are three risk states: “normal”, “warning” and “hazard” (Fig. 9). Table 3 presents the created knowledge base. Within the considered solution, a risk state is determined by means of the Larsen method. Certainly, the accuracy of the overall presented setting depends heavily on the quality of forecasts delivered by the prediction module.

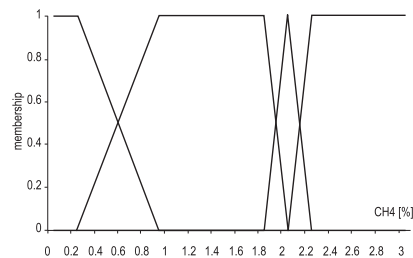


Fig. 8. Fuzzy set representation of methane concentration dynamics.

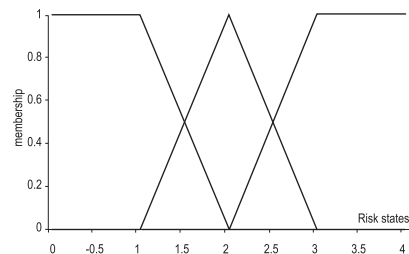


Fig. 9. Fuzzy set representation of the risk states.

Table 3

The knowledge base connecting risk states with methane concentration.

Rule	Methane concentration	Dynamics of methane changes	Risk state
1	Normal	No changes	Normal
2	Normal	Increasing	Normal
3	Normal	Quickly increasing	Warning
4	Admissible	No changes	Warning
5	Admissible	Increasing	Warning
6	Admissible	Quickly increasing	Hazard
7	Boundary	–	Hazard
8	Exceeded	–	Hazard

7. Experimental validation of our data flow assumptions

In Sections 3–6 we focused on the most important architectural aspects of the proposed data-driven decision support system. Now let us proceed with the first step of experimental validation of the assumptions that we made during the system design. The goal of this section is to present an example of a data mining investigation which illustrates our system performance when handling real-life problems related to coal mining processes. The considered task is to construct a model capable of predicting dangerous concentrations of methane at longwalls of a coal mine. The three data sets used in our experiments – reported below and in Section 9 – were obtained from active Polish coal mines.

In the first experiment we considered the data from the period between March 2, 2014 and June 16, 2014. It included readings from 28 different sensors. Since the data was arranged into records describing 10-min periods sampled once per second, the data consisted of 16,800 numerical features, i.e., 600 values per sensor. Therein, we used our ETL2 layer just to clean the data and create suitable dependent variables. Their values indicated whether a warning threshold was reached in a period between three and six minutes after a given measurement, for selected methane meters denoted as MM263, MM264 and MM256 (see Fig. 10), located in the most exposed areas of the monitored longwalls. If a given data row corresponded to a time period between t_{-599} and t_0 , then its dependent variable value for a meter MM was “warning”, if and only if $\max\{MM(t_{181}), \dots, MM(t_{360})\} \geq \rho$, where ρ is a safety threshold. The value of this threshold may vary for different longwalls, however, it is usually set between 1% and 1.5%. This experiment did not include a step of window-based feature extraction, so data records were composed of raw sensor readings arranged in 10-min time series, with measurements taken every second.

Based on such obtained data, we organized an open data mining competition at one of international conferences [23]. We used our own on-line competition platform² with rules comparable to other popular initiatives in the field. The data set in the competition was prepared in a tabular format, as two separate subsets – the training data set and the testing data

² <https://knowledgepit.fedcsis.org/contest/>

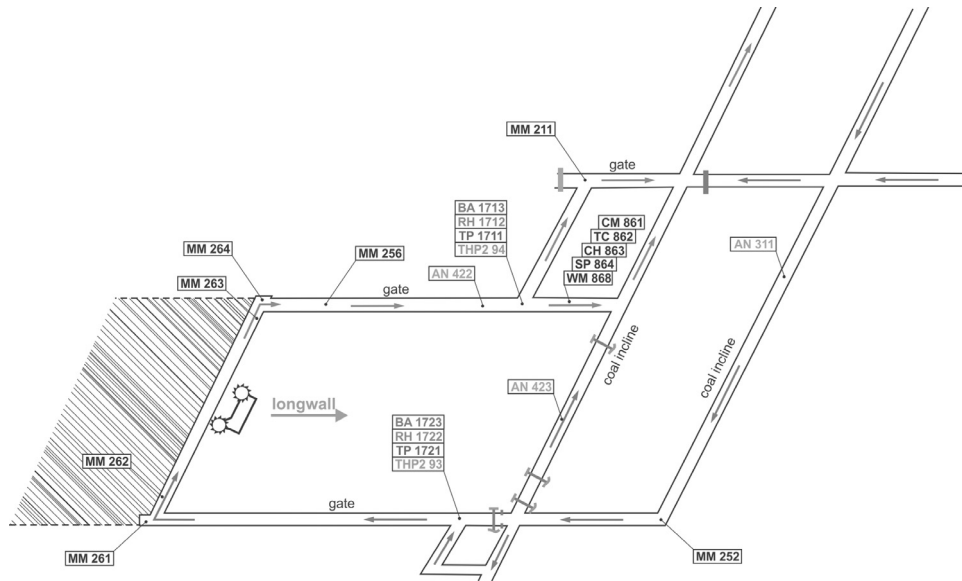


Fig. 10. A scheme of the mining process corresponding to the data set considered in [23].

Table 4

Preliminary and final results of the top-ranked teams in the competition [23].

Rank	Method	Preliminary	Final score
1	Zagorecki [49]	0.96662590	0.95926715
2	Boullé [7]	0.94607103	0.94392893
3	Ruta and Cen [45]	0.93371596	0.94369948
...
6	Pawłowski and Kurach [39]	0.96853101	0.94002446
...
...	Baseline	0.89302460	0.90044912

set. The training set contained sensor readings registered within 51,700 time periods. Those periods were overlapping and organized in a chronological order. The periods in the testing set did not overlap and they were given in a random order. Values of dependent variables were provided to the participants only for the training set. During the competition, for each of submitted solutions, preliminary classification results were calculated for an unknown sample of the testing set. Final results were calculated over the complete testing set.

The participants were also provided with the mining scheme in Fig. 10, so the placement of all sensors represented in the considered data set was known. This scheme, combined with the provided description of each sensor (following our sensor card standard), constitutes an important source of domain knowledge which makes it possible to understand potential dependencies between readings of different sensors. In this particular example, the cutter loader moves along the longwall between sensors MM262 and MM264. The bigger are the currents consumed by the cutter loader, the more efficient is its mining work which, in theory, results in more methane emitted to the air. The arrows on the provided scheme show the directions of the air flow (along with the methane flow) in the corridors. If a methane concentration measured by any of the sensors reaches an alarm level, the cutter loader is switched off automatically. However, if we were able to predict ahead the warning methane concentrations, we could reduce the speed of the cutter loader early enough to give the methane more time to spread out, before the necessity of switching off the whole production line [36].

The quality criterion chosen to evaluate the competition submissions was based on the Area Under ROC Curve (AUC) measure. It was selected because of a sparsity of positive examples for each of the considered dependent variables. The AUC values were computed independently for each of three considered sensors. The score received by a given submission was equal to the mean of the obtained AUC values. Table 4 shows the scores of top-ranked teams together with the score obtained by a baseline model created by averaging 10 simple rule-based classifiers computed using the aforementioned *RoughSets* package.

Our goal was to find out whether its most successful participants might use data processing techniques similar to those that we considered in the previous sections. In our opinion, the most interesting outcome was that a vast majority of solutions followed the ideas of producing sliding window aggregations and that such aggregations – treated as low-level features – were useful while learning various prediction models. The winning approach [49] assumed generation of a large number

of variables characterizing sensor measurements and operating with the time series derived from those measurements. Separate random-forest-based models were then created to predict the “warning” states for each of three considered methane meters. The second-best solution [7] focused on a problem of distribution drift between the training and testing data sets. Informativeness of each considered feature with respect to both classification and drift detection was evaluated. As a result, the training data set was reduced to a single sensor per target class. The prediction model was then generated by the Naïve Bayes classifier. The third-best method was based on a logistic regression model computed over a small subset of sensor observations [45]. The authors utilized their self-organizing framework to choose this particular model out of a number of other solutions including decision trees, support vector machines, etc. From our perspective, that was an additional inspiration to consider relatively simple models relying on filter-based feature selection. Among other prominent approaches used by participants of the competition, there were also deep learning models using the LSTM networks [39]. However, in this case the final results turned out significantly worse than the preliminary score.

By publishing this data set and defining the corresponding problem in the form of a competition task, we were able to obtain and analyze a large number of solutions submitted by participants. Altogether, these solutions can be regarded as the current state-of-the-art in the predictive analysis of multivariate time series data. We were able not only to use the competition results as a reference in our own research outlined in Sections 8 and 9 but also to incorporate some of the innovative ideas to improve methods which are now available in our system. Based on the analysis of the most successful solutions submitted to the competition, we reached a conclusion that a robust prediction of methane concentration levels can be achieved even when a small subset of features is used for constructing the model, though it is certainly a non-trivial task to find out features that are the best ones for this purpose. Although the solution that obtained the highest evaluation score used nearly 5000 features in the learning process, a few of the other top-ranked teams achieved similar results with models considering far fewer features. Moreover, as already mentioned, many top teams used quite similar techniques of feature extraction to those discussed in Section 5. We treat it as a kind of empirical validation of our assumptions made while designing and implementing the DISESOR system.

8. Feature selection and feature subset ensembles

Feature selection is a typical task in knowledge discovery and data analytics [9,10]. In the case of designing decision support systems, the scope of feature selection is twofold, related to both interaction with domain experts and analysts while running a given system on-line, as well as off-line exploration of the gathered data in order to find out the best algorithms and prepare the best possible feature sets for further processing. It is herein worth noting that the step of feature selection – conducted independently or in an iterative fashion – is often taken into account while building systems aimed at equipment and environment monitoring in coal mines, in combination with various machine learning methods such as neural networks, support vector machines, etc. [8,33].

With regard to the architecture discussed in this paper, the requirement of feature selection is quite straightforward, since a large number of window-based ETL2-driven features could increase the time needed to train, use and adapt forecasting models. For problems related to coal mining, as it could be already seen in Section 7, the amount of generated features can reach several thousands depending on how the feature extraction mechanism is configured. An excessive amount of created features provides a great potential for data-driven reasoning but, on the other hand, many of those features may be redundant or others could be irrelevant from the point of view of a given problem.

Following the outcomes of the previously-discussed competition, we decided to focus on filter-based methods which (comparing to wrapper and embedded techniques) assure relatively high computational efficiency, as well as independence of the resulting feature sets from a particular model. This last property allows the obtained feature sets to be used in combination with various types of forecasting approaches. Consequently, referring to the overall system layout displayed in Fig. 1, the feature selection algorithms are developed within the analytical module while their outcomes – after consultation with the users – can be utilized to build models that are plugged into the prediction module.

Among filter-based feature selection methods, we pay special attention to multivariate algorithms. Such approaches rely on inter-feature dependencies when selecting a feature subset. Most of multivariate filtering algorithms attempt to avoid including unnecessary features by measuring redundancy within the selected subset. As a result, they produce quite compact feature sets. This is a big advantage in the context of applications of our system. As mentioned in Section 2, one of the most prominent examples of this approach are methods based on the mRMR framework [34,40]. Such methods iteratively select features that provide the most relevant information regarding dependent variable values (e.g.: are highly correlated or have a high value of mutual information index) and, on the other hand, are less dependent on the already-selected features. A pseudo-code of this approach is presented as Algorithm 1.

In DISESOR we implemented a novel modification of standard mRMR approach. Comparing to the original version, our changes are related to the criteria for selecting the best feature in each iteration and to a stopping condition of the algorithm. First, we select a feature that maximizes the difference between its relevance (the dependency score $\phi(a, d)$) and its maximal dependency on features selected before. Second, we stop the algorithm if the feature selected in a given iteration does not pass the random probe test, i.e., the estimation of the probability that a randomly generated feature obtains a higher score than the selected feature exceeds an allowed threshold. Thus, we guarantee compactness and the relatively high independence of the resulting feature set.

Another already-mentioned approach implemented in our system refers to computation of decision reducts developed within the theory of rough sets [38]. As briefly outlined in Section 2, a decision reduct can be defined as an irreducible subset of features that provides sufficient information to distinguish data records having different values of a dependent variable. Precisely, we followed the approach called Dynamically Adjusted Approximate Reducts (DAAR), where a statistical test based on random probes (analogous to the test embedded into Algorithm 1) is used to avoid selection of features that are likely to distinguish data records supporting different target classes only by chance [24].

Implementation of DAAR is available in *RoughSets* [44] – the R System package including rough-set-based machine learning techniques. It is worth mentioning that such techniques gain an increasing attention in applications [26]. We used a variant which lets generate an *ensemble* of different and possibly diversified feature subsets. Let us note that ensemble-based feature selection approaches are quite popular, although their final goal is rather to produce a single feature set as a combination of particular ensemble components [2]. On the other hand, our DAAR-based approach is to maintain a collection of not combined feature sets, train forecasting models over each of them separately and then treat them as a kind of ensemble of classifiers [42]. Basing on our research, we observe that diversity of feature sets is one of the aspects leading towards robustness of classifier and/or regression ensembles [22]. Moreover, such understood idea of operating with feature subset ensembles can be combined with some meta-learning methods that proved to work efficiently in the considered field of study [30].

Comparing to Algorithm 1, the DAAR's pseudocode differs in two aspects. First, a dependency function ϕ is employed to measure a newly added feature \tilde{a} with respect to d and the whole current set A' rather than its particular elements. Second, at the very end there is an additional phase of eliminating from A' any redundant features. Actually, this particular stage whose aim is to produce possibly smallest feature subsets makes the rough-set-inspired algorithms quite unique comparing to other feature selection methods.

By default, we compute 10 DAAR-based feature subsets. In order to diversify them, we can first use a bootstrapping method with a stratified data sampling. Next, we discretize the data using the maximum local discernibility method implemented in the above-mentioned *RoughSets* package. Then, we use the default implementation of DAAR configured so as to randomly examine 10% out of the total number of available features. Moreover, diversification of the final feature subsets is boosted by following randomly generated permutations of elements of A' in the elimination phase – a technique that is implemented in the considered package as well.

In [19], it is further proposed to randomly divide the training cases – represented by ETL2-driven feature vectors – into two disjoint groups: group A used for a purpose of DAAR-based feature selection and group B used for learning regression models. Some of feature sets resulting from DAAR are merged, so the number of elements in an ensemble is kept reasonably low. Regression models can be calculated using different algorithms, including regression trees, SVM regressor, generalized linear model available in R, etc. [31]. The ensemble is incrementally extended with a new model, if its results over the group B are significantly different from any of the previously added models.

9. Experimental validation of feature selection methods

In the second part of our experiments, we considered exactly the same sensor readings as those used in the first one, but now taking into account our feature extraction procedure. We also included into our analysis the two remaining data sets with a different number of sensors involved. In the new data sets, each time period is described by a number of statistics (like those listed in Table 2) computed over consecutive time windows. We applied two feature selection methods discussed in Section 8 and we examined the AUC scores of prediction outcomes obtained using simple (ensembles of) regression models. As a reference, we used a baseline model with no feature selection.

Let us remind that in the proposed system the feature selection algorithms work on historical data preprocessed using ETL2. Their outputs are then utilized during off-line construction of forecasting models. These models need to be simple enough to apply them for on-line prediction based on streams of sensor readings. This is our main reason to focus on relatively basic models. On the other hand, we need to verify whether such models do not cause a significant decrease of prediction quality comparing to more advanced solutions. This is why we additionally compared the implemented feature selection methods to the results obtained using the aforementioned competition winner approach.

First, we applied our version of mRMR (see Algorithm 1). It was used separately for each of three dependent variables. We obtained three small subsets of features containing three, six and seven elements, respectively. With these subsets, we trained three independent logistic regression models (no regularization was employed). Although we used a very simple prediction method and a fully automated feature selection, the average AUC of this solution for the final testing set was 0.9413. This result would give us the sixth place in the competition ranking. It is also worth mentioning that this score was lower than that achieved by the team ranked in the second position by only 0.0026.

In order to verify whether our custom design of mRMR stopping condition is suitable for the considered task, we repeated the experiment with the probe condition switched off. Fig. 11 shows the results of the forecasting model trained using features selected in 25 consecutive iterations of Algorithm 1. It can be seen that for each of the considered dependent variables the results obtained using our stopping criterion were close to optimal. Moreover, by virtue of the random probe test, the resulting feature subsets were compact. Overall, they contained the lowest number of features among the solutions submitted by the top-ranked teams in the competition.

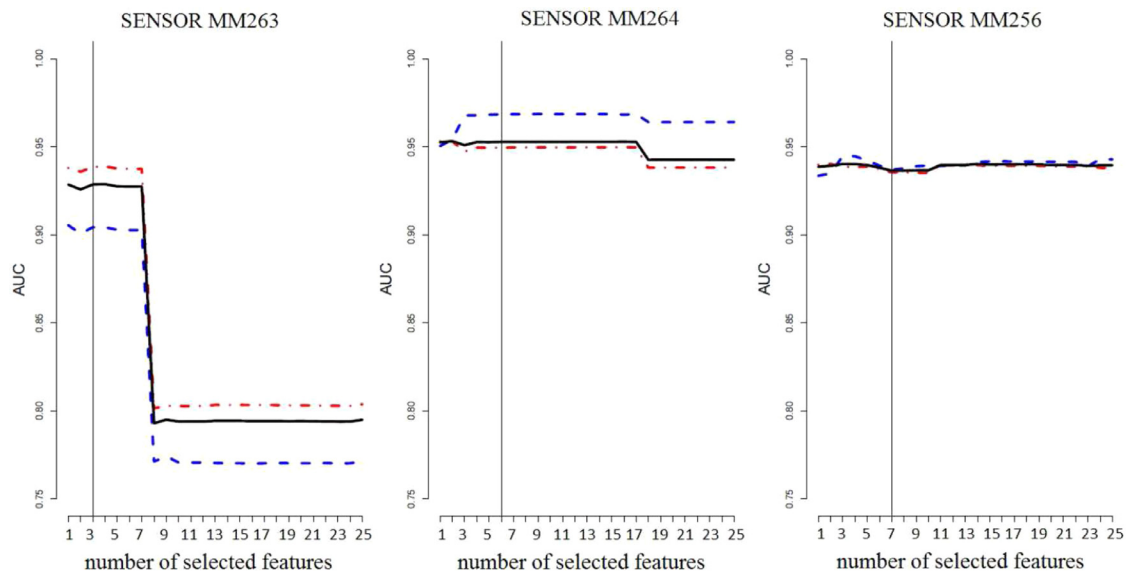


Fig. 11. The AUC values obtained for linear regression models trained on features selected in subsequent iterations of Algorithm 1 for each of dependent variables in the competition data: dashed blue lines show the scores on the preliminary testing set; dashed–dotted red lines show the scores on the final testing set; thick black lines show the scores on the joined testing set; thin vertical lines mark the iteration on which the stopping condition was triggered. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5

Logistic regression performance for the competition data set [23], in combination with the implemented feature selection methods, compared to the competition winner.

	All features	mRMR subset	DAAR subsets	Zagorecki [49]
Preliminary	0.8111	0.9397	0.9417	0.9666
Final	0.8129	0.9413	0.9545	0.9592

We also tested a simple mode of DAAR-based approach outlined in Section 8. For each of the dependent variables, we computed 10 diversified feature subsets. Then, for each of them, we computed a logistic regression model using the same algorithm as for features selected with mRMR. Finally, we created a simple ensemble by averaging the models' predictions. A comparison of the results obtained for these two feature selection methods and the case with no feature selection used at all is shown in Table 5. The final score achieved by the ensemble of DAAR-based regression models was 0.9545 which would give it the second rank in the competition.

In order to thoroughly investigate the performance of the compared models, we tested them on two additional data sets that were obtained from different coal mines. The first of those sets contained sensor readings from over a month (November 2007–December 2007). Similarly as the first set, there were three target methane meters (MM532, MM533 and MM534) and the sensor reading frequency was one per second. The data set contained readings from eight sensors plus additional information regarding the coal shearer status. After the initial preprocessing, the data consisted of 51,329 records corresponding to 10-min periods of sensor readings. These records were divided into two disjoint sets – one for training the compared models and the second one for validation. The testing set corresponded to the last two weeks of sensor readings.

In the second of the considered new data sets, there was only one target methane meter (denoted by MWR116 in Fig. 12) but the data spanned across five months (September 2013–January 2014). In this set, there were readings from 14 sensors, sampled once per minute. After the initial preprocessing, the set consisted of 204,465 records. They were divided into separate training and testing sets as well. The test period corresponded to the last two months of sensor readings. The data sets used in our experiments are available on-line.³ The results obtained for each of the target sensors from all three data sets are presented in Table 6. In addition to our own models, we include there the results obtained using our own implementation of the model reported by the competition winner.

Table 6 confirms the level of accuracy that can be obtained using our approach, taking into account its subsequent layers of data integration, data preprocessing and finally the forecasting model construction, following the principles of feature subset ensemble selection. It is worth noticing that both feature selection approaches – the more standard mRMR and the ensemble-oriented DAAR – yield good results, even when combined with one of the simplest possible prediction techniques

³ http://adaa.polsl.pl/data_and_software.html

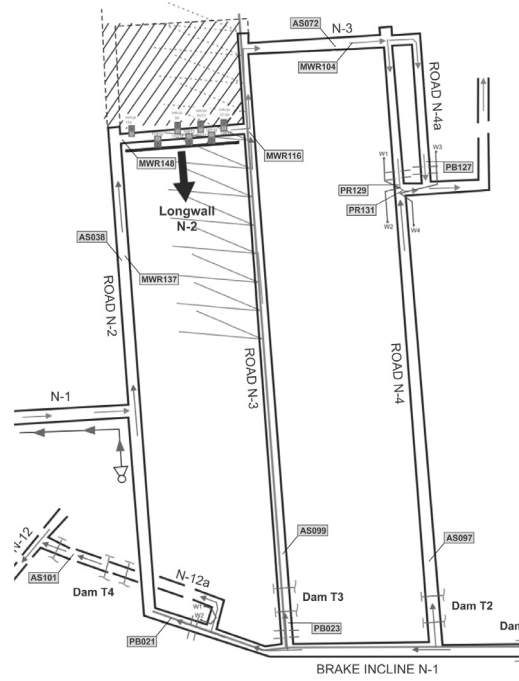


Fig. 12. A scheme of the mining process corresponding to the second considered data set.

Algorithm 1: Implemented version of mRMR method.

Input: set of features A and dependent variable d ;
 $\phi : A \times A \cup \{d\} \rightarrow R^+$ function for measuring dependency;
 $N \in \mathbb{N}$; $\varepsilon \in [0, 1)$;

Output: subset of features $A' \subseteq A$

```

1 begin
2   stopFlag  $\leftarrow$  FALSE;
3    $A' \leftarrow \{\arg \max_{a \in A} \phi(a, d)\}$ ;
4    $A \leftarrow A \setminus A'$ ;
5   while stopFlag == FALSE do
6      $\bar{a} \leftarrow \arg \max_{a \in A} (\phi(a, d) - \max_{b \in A'} \phi(a, b))$ ;
7     foreach  $i \in 1, \dots, N$  do
8        $\bar{p}_i \leftarrow$  random permutation of  $\bar{a}$ ;
9     end
10    if  $\frac{|\{i: |\phi(\bar{p}_i, d)| > |\phi(\bar{a}, d)|\}| + 1}{N+2} > \varepsilon$  then
11      stopFlag  $\leftarrow$  TRUE;
12    else
13       $A' \leftarrow A' \cup \{\bar{a}\}$ 
14    end
15  end
16 end

```

– the logistic regression. From the prediction accuracy perspective, they perform comparably to the model developed by the competition winner which was manually tuned for over two months. Furthermore, our models are easy to maintain and efficient to compute. Moreover, they are understandable for the system users and domain experts by means of operating with small subsets of intuitively defined features. In particular, this is why the DAAR-based method introduced in [24] and further extended in [19] is now provided as one of operators in our prediction module.

Table 6

A comparison of logistic regression performance (AUC values) for individual target methane meters from the three data sets considered in our study.

Target	mRMR subset	DAAR subsets	Zagorecki [49]
MM256	0.9432	0.9579	0.9439
MM263	0.9374	0.9564	0.9760
MM264	0.9433	0.9492	0.9579
MM532	0.9176	0.9170	0.8968
MM533	0.8501	0.8681	0.8283
MM534	0.9276	0.9321	0.9299
MWR116	0.9389	0.9431	0.9575

It is also worth to notice that the DAAR-based algorithm achieved the highest average AUC on all seven target sensors (0.9320 vs. 0.9272 in the case of the competition winner). Although the paired Wilcoxon test did not revealed statistically significant differences in the results, it is sufficient to conclude that it can successfully replace more complicated and hardly interpretable machine learning approaches. Moreover, computation time required to train our model was an order of magnitude lower. For instance, for the third data set, our model was constructed in 19 min, whereas the construction of the random forest model took nearly five hours.

10. Conclusions

In this paper, we referred to the decision support system that we designed and implemented for the purposes of monitoring the coal mine processes and predictive analytics over streams of sensor readings. We dived into construction and maintenance of natural hazard forecasting models, as well as the system modules and functionalities responsible for data acquisition and storage, data cleaning and transformation. Our contributions correspond to adjusting the layers of our system to the requirements of multi-sensor data exploration and introducing very simple though powerful methods for feature extraction and selection. This paper continues the previous research on both general architecture and specific machine learning methods used in our system [25,36]. However, a number of important aspects, such as, e.g., sensor data integration (Section 4 and Appendix A) and hybridization of ensemble-based feature selection techniques and regression models (Sections 8 and 7), are fully new.

The above novelties are embedded into a modular system architecture whose particular components can be easily modified or enriched. For example, at a data repository level, we utilized the open source PostgreSQL software which could be, if necessary, replaced by some other database solutions and configurations. Furthermore, for data flow and user interaction purposes, we adapted and extended the open RapidMiner environment that includes a number of useful data preprocessing and visualization operators [21]. Finally, with regard to the data mining algorithms, we took an advantage of some already existing, relatively simple methods and libraries [44], rather than developing new advanced methodologies that would be hard to control within such a dynamic decision support system environment.

For the architecture validation purposes we used techniques that are available in our system to cope with a practical task of methane concentration forecasting. We organized an international data mining competition using one of available on-line competition platforms [23] and we compared our own methods to the best submitted solutions [49]. The outcomes show that our approach is fully competitive and simple enough to embed into the prediction module designed within the considered system. The competition also confirmed usefulness of techniques of sliding-window-based processing of the streaming data. As a result, we decided to implement in our system a set of predefined functions providing aggregated features of a data stream in a given time window.

In future, we will extend our framework to facilitate deeper interaction between the system users and operations that can be performed within our analytical module. From the perspectives of feature selection and predictive modeling, the system requires iterative, interactive and adaptive methods that can adjust to user recommendations and preferences in the context of the analyzed data. We believe that model maintenance and user interaction should be implemented at the level of particular components of forecasting ensembles, e.g., by further diversification of feature subsets used in particular models and reporting to users their performance within an ensemble [24]. We are also going to develop different levels of interactive feature selection, e.g., by conducting the algorithms at level of collections of parameters extracted from given sensors and time frames instead of single features [20].

Finally, we will work on extending and better utilizing information registered while adding new data sources. There are a number of places where such information – included in our sensor cards – could be useful to configure the steps of consistent data gathering, data preprocessing and learning forecasting models. One of yet unexplored possibilities is to use it also to customize window-based and inter-window-based aggregations applied for particular sensors and groups of sensors during the process of feature extraction. Indeed, it is known that the stage of feature extraction should take into account semantics of both the overall forecasting task and particular inputs that may help to build its solution [10,50]. In the case of multi-sensor analytics in the domain of underground coal mining, it may mean applying different feature

extraction strategies for different types of sensors, as well as constructing higher-level features basing on spatial information represented by the corresponding mining schemes, such as the ones shown in Figs. 10 and 12.

Acknowledgment

This research was supported by the Polish National Centre for Research and Development (NCBiR) grant PBS2/B9/20/2013 in the frame of the Applied Research Programme.

Appendix A. Sensor card specification – an example of contents that would be registered for a methane meter

#	Specification item description	Content stored in sensor card
	A. Basic information	
1.	Measured value	e.g.: concentration
2.	Measured value's name in the system	e.g.: methane concentration
3.	Name and type of sensor/device	e.g.: methane meter MM-4
4.	Location	
	a) Mine face (blind drift)	e.g.: yes
	b) Longwall (sporadically)	e.g.: yes
	c) Top road	e.g.: yes
	d) Bottom road	e.g.: yes
	e) Cross heading	e.g.: yes
	f) Incline drift	e.g.: yes
	g) Raise	e.g.: yes
	h) Cross-cut	e.g.: yes
	i) Chamber	e.g.: yes
	j) Machine (longwall shearer, conveyor...)	e.g.: no
	k) Device (switchgear, transformer...)	e.g.: yes
	...other (specify)	
5.	Identification of data source	
	a) Sensor/measurement device type	e.g.: not applicable
	b) The smallest functional subassembly of a monitored device that the sensor/measurement device is connected with	e.g.: not applicable
	c) Hierarchy of higher level subassemblies	e.g.: not applicable
6.	Purpose	
	a) Safety	e.g.: yes
	b) Devices that run monitoring	e.g.: no
	c) Production monitoring	e.g.: no
	d) Production preparation	e.g.: no
	...other (specify)	
7.	Name(s) of system(s) transmitting the data from sensor/device	
	a) System trade name and denotation	
	b) System producer	
	c) Trade name and denotation of a system of higher order that visualizes the data	
	d) Higher level system producer	
8.	Name and code of sensor/device in the system	
	a) Name	e.g.: methane meter 283 (three digit number given by the coal mine services)
	b) Denotation in the system	e.g.: MM283
	c) Measurement system	e.g.: dispatching System SMP-NT
9.	Categorization of sensor/device	
	a) Mine atmosphere parameters (climate and gas hazards)	e.g.: no
	b) Hard coal quality	e.g.: no
	c) Electric network parameters	e.g.: no
	d) Parameters of devices working	e.g.: no
	e) Media consumption forecasting	e.g.: no
	f) Seismic hazards	e.g.: no
	g) Methane hazards	e.g.: yes
	h) Device diagnostic states	e.g.: no
	i) Fire hazards	e.g.: yes
	j) Water hazards	e.g.: no
	k) Bumps hazards	e.g.: no
	l) Breakout hazards	e.g.: no
	m) Coal dust blow-up hazards	e.g.: no
	n) Landslide hazards	e.g.: no
	o) Eruption hazards	e.g.: no
	p) Hydrogen sulfide hazards	e.g.: no
	q) Radioactive substance hazards	e.g.: no
	...other (specify)	

- B. Metrological information**
- 10. Measurement range and value unit**
 a) Measured value range e.g.: 1. $0 \div 5\% CH_4$ 2. $5 \div 100\% CH_4$
 b) Measured value unit e.g.: $\% CH_4$
- 11. Measured value unit scaler**
 e.g.: no data
- 12. Basic technical data**
 a) Sensor/device measuring accuracy e.g.: 1. $\pm 0.1\% CH_4$ for concentrations from the range 0 – 2% CH_4 2. $\pm 5\% rdg$ (reading) for concentrations from the range 2 – 5% CH_4 3. $\pm 3\% CH_4$ for concentrations from the range 5 – 60% CH_4 4. $\pm 5\% rdg$ (reading) for concentrations from the range 60 – 100% CH_4
 b) Transmission resolution e.g.: up to 2 decimal digits
 c) Measurement storage resolution e.g.: up to 2 decimal digits
 d) Writing format e.g.: xx.xx
- 13. Device type**
a) Device
 1. Measuring instrument e.g.: yes
 2. Sensor e.g.: no
 3. Detector (detects a value without its measuring) e.g.: no
b) Measuring instrument/sensor/detector
 1. Electric e.g.: no
 2. Electronic e.g.: yes
 3. Pointing e.g.: yes
 4. Recording e.g.: yes
 5. Analog e.g.: no
 6. Digital e.g.: yes
 7. Discrete/binary (contact output signal, 0/1) e.g.: no
 8. Single-range e.g.: no
 9. Multi-range e.g.: yes
 10. Single-functional e.g.: yes
 11. Multi-functional e.g.: no
- 14. Types of functionalities (realized independently or in the system)**
 a) Registering e.g.: depending on the system settings
 b) Registering and signalling e.g.: yes
 c) Registering and alarming e.g.: depending on the system settings
 d) Registering and controlling e.g.: yes
- 15. I/O type of sensor/device**
a) Sensor/device inputs
 1. AI: analog input e.g.: no
 2. DI: discrete (binary) input e.g.: two-state, cooperating with non-voltage contacts; line state monitoring
 3. CU: analog summing input e.g.: no
 4. CD: analog subtracting input e.g.: no
 5. MI: discrete unordered input e.g.: no
 6. MMI: discrete ordered input e.g.: no
b) Sensor/device outputs
 1. AO: output e.g.: no
 – Voltage e.g.: not applicable
 – Current e.g.: not applicable
 – Frequency e.g.: not applicable
 2. DO: discrete (binary) output e.g.: yes
 – Active e.g.: no
 – Passive e.g.: two, with independently set thresholds, separated galvanically from the meter
 – Resistant e.g.: no
 3. CO: impulse (counting) output e.g.: no
 4. ESI: type of communication interface and protocol (external system interface) e.g.: data transmission using feeder cable
 5. MO: discrete unordered output e.g.: no
 6. MMO: discrete ordered output e.g.: no
 ...other (specify)
- C. System information**

- 16. The type of collected data**
- a) Device state e.g.: no data
 - b) Name of sensor/device and its denotation in the system e.g.: *Mxxx* or *MMxxx*
 - c) Location of sensor/device in a mine e.g.: yes
 - d) Value of a measured variable e.g.: yes
 - e) Two-state outputs state e.g.: no data
 - f) Controlling outputs states e.g.: no data
 - g) Signalling outputs states e.g.: no data
 - h) Alarming outputs states e.g.: no data
 - i) Date of measurement e.g.: yes
 - j) Time of measurement e.g.: yes
 - k) Name of events e.g.: no data (inserted manually)
 - l) Date of events e.g.: no data
 - m) Time of events e.g.: no data
 - ...other (specify)
- 17. Measurement quality criteria**
- a) Sensor/device range e.g.: $0 \leq \% CH_4 \leq 100$ negative values (-0.1% , -0.2%) acceptable and interpreted as the result of unclearing the detector
 - b) Overflow e.g.: $\% CH_4 < 0$ or $100 < \% CH_4$
 - c) Wrong measurement (incorrect, etc.) e.g.: $\% CH_4 < -0.2$ or $100 < \% CH_4$
 - d) Lack of measurement e.g.: no data, independent from the cause; in the monitoring system: – The sensor / device symbol becomes gray; – The value is replaced by “??” – Missing values are not filled
- 18. Identification of sensor/device state**
- a) Configuration e.g.: no data
 - b) Programming e.g.: no data
 - c) Calibration e.g.: during the calibration measured values are sent to the monitoring system with the additional feature informing that the sensor is in the calibration phase
 - d) Break of communication e.g.: no data
 - Possibility to identify the working state with:
 - e) Characteristic data (values) e.g.: not applicable
 - f) Characteristic chart (constant) e.g.: not applicable
 - g) Duration (calibration, configuration) e.g.: not applicable
 - ...other (specify)
- 19. Measurement quality criteria due to the change dynamics**
- a) Probably e.g.: due to possibility of events with very high dynamics of changes (ejections, other abrupt methane outflows) there is no range of changes defined, irrespective of the sampling time
 - b) Improbably e.g.: no
- 20. Threshold for significance of measurement changes**
- a) Non-significant changes e.g.: below $\pm 0.1\% CH_4$
 - b) Significant changes e.g.: over $\pm 0.1\% CH_4$
- 21. Measuring frequency and conditions**
- a) Type of sampling e.g.: continuous sampling
 - b) Data fetching frequency e.g.: 2 s
 - c) Are all the data transmitted to the system e.g.: no – only changing values are registers on the first level (repeating measurements are not written)
 - d) Criteria of qualifying measurements to be transmitted and/or registered in the system e.g.: no data

22.	Predefined measurement limits a) Names, symbols of limit values for: 1. Warning threshold(s)	e.g.: yes e.g.: WT (warning threshold) – configurable e.g.: AT (alarm threshold) – configurable
	2. Alarm threshold(s)	
	3. ...other (specify)	
	b) Measured value state names:	e.g.: yes e.g.: \leq WT e.g.: between WT and AT e.g.: \geq AT
	1. Normal state	
	2. Warning state	
	3. Alarm state	
	...other (specify)	
23.	Sensor/device working mode	
	a) Continuous	e.g.: yes
	b) Short-duration	e.g.: no
	c) Discontinuous	e.g.: no
	d) Seasonal (irregular, between switching on and off)	e.g.: no
24.	Measurement timestamp	
	a) Time (timestamp)	e.g.: yes
	1. Measuring	e.g.: no
	2. Disc/database storage	e.g.: yes (in the telemetry center)
	...other (specify)	
	b) Time writing system	e.g.: yes
	1. Local (system)	e.g.: yes
	2. GMT	e.g.: no
	3. Central European	e.g.: yes (one-hour gap in data after changing to Summer time; one hour of duplicated data after changing to Winter time)
	...other (specify)	
25.	Measured value	
	a) Absolute	e.g.: yes
	b) Relative	e.g.: no
	c) Conventional	e.g.: no
26.	Recommended aggregation time	
	a) Required (recommended) measurement aggregation time that reflects the value of measured variable in specified time interval	e.g.: 24 h
	b) Aggregation method	
	1. Average over the aggregation time	e.g.: yes
	2. Average over the measurements' number	e.g.: no
	...other (specify)	
27.	Additional information about sensor/device	
	a) Sensor/device is	
	1. Autonomic	e.g.: yes
	2. Related to another sensor(-s)	e.g.: no
	b) Are there any dependencies or correlations between related sensors?	e.g.: no
28.	Additional description (not included in previous questions)	
	What else should be known about a given sensor/device?	

References

- [1] A. Abecker, T. Brauer, B. Magoutas, G. Mentzas, N. Papageorgiou, M. Quenzer, A sensor and semantic data warehouse for integrated water resource management, in: *Proceedings of the 2014 EnviroInfo*, 2014, pp. 509–516.
- [2] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, *Bioinformatics* 26 (3) (2010) 392–398.
- [3] C.C. Aggarwal, *Managing and Mining Sensor Data*, Springer, 2013.
- [4] A. Appice, P. Guccione, D. Malerba, A. Ciampi, Dealing with temporal and spatial correlations to classify outliers in geophysical data streams, *Inf. Sci.* 285 (2014) 162–180.
- [5] R. Bakirov, B. Gabryś, D. Fay, Multiple adaptive mechanisms for data-driven soft sensors, *Comput. Chem. Eng.* 96 (C) (2017) 42–54.
- [6] R. Barton, *Talend Open Studio Cookbook*, Packt Publishing, 2013.
- [7] M. Boullé, Prediction of methane outbreak in coal mines from historical sensor data under distribution drift, in: *Proceedings of the 2015 RSFDGrC*, 2015, pp. 439–451.
- [8] M. Cerrada, R.V. Sánchez, D. Cabrera, G. Zurita, C. Li, Multi-stage feature selection by using genetic algorithms for fault diagnosis in gearboxes based on vibration signal, *Sensors* 15 (9) (2015) 23903–23926.
- [9] G. Chandrashekar, F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.* 40 (1) (2014) 16–28.
- [10] P. Domingos, A few useful things to know about machine learning, *Commun. ACM* 55 (10) (2012) 78–87.
- [11] F. Dong, G. Zhang, J. Lu, K. Li, Fuzzy competence model drift detection for data-driven decision support systems, *Knowl.-Based Syst.* 143 (2018) 284–294.
- [12] C.D. Duan, Z.D. He, H.K. Jiang, A sliding window feature extraction method for rotating machinery based on the lifting scheme, *J. Sound Vib.* 299 (4–5) (2007) 774–785.
- [13] B. Erinle, *JMeter Cookbook*, Packt Publishing, 2014.
- [14] L. Fortuna, S. Graziani, A. Rizzo, M.G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*, Springer Science & Business Media, 2007.

- [15] T.C. Fu, A review on time series data mining, *Eng. Appl. Artif. Intell.* 24 (1) (2011) 164–181.
- [16] J.A. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, *ACM Comput. Surv.* 46 (4) (2014) 44:1–44:37.
- [17] C. Goumopoulos, B. O'Flynn, A. Kameas, Automated zone-specific irrigation with wireless sensor/actuator network and adaptable decision support, *Comput. Electron. Agric.* 105 (C) (2014) 20–33.
- [18] T. Grychowski, Hazard assessment based on fuzzy logic, *Arch. Min. Sci.* 53 (4) (2008) 595–602.
- [19] M. Grzegorowski, Massively parallel feature extraction framework application in predicting dangerous seismic events, in: *Proceedings of the 2016 FedCSIS*, 2016, pp. 225–229.
- [20] M. Grzegorowski, A. Janusz, D. Ślęzak, M.S. Szczuka, On the role of feature space granulation in feature selection processes, in: *Proceedings of the 2017 Big Data*, 2017, pp. 1806–1815.
- [21] M. Hofmann, R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*, Chapman & Hall/CRC, 2013.
- [22] A. Janusz, M. Grzegorowski, M. Michalak, L. Wróbel, M. Sikora, D. Ślęzak, Predicting seismic events in coal mines based on underground sensor measurements, *Eng. Appl. Artif. Intell.* 64 (2017) 83–94.
- [23] A. Janusz, M. Sikora, L. Wróbel, S. Stawicki, M. Grzegorowski, P. Wojtas, D. Ślęzak, Mining data from coal mines: IJCRS'15 data challenge, in: *Proceedings of the 2015 RSFDGrC*, 2015, pp. 429–438.
- [24] A. Janusz, D. Ślęzak, Computation of approximate reducts with dynamically adjusted approximation threshold, in: *Proceedings of the 2015 ISMIS*, 2015, pp. 19–28.
- [25] A. Janusz, S. Stawicki, M.S. Szczuka, D. Ślęzak, Rough set tools for practical data exploration, in: *Proceedings of the 2015 RSKT*, 2015, pp. 77–86.
- [26] Y. Jing, T. Li, H. Fujita, Z. Yu, B. Wang, An incremental attribute reduction approach based on knowledge granularity with a multi-granulation view, *Inf. Sci.* 411 (2017) 23–38.
- [27] J. Kabiesz, Effect of the form of data on the quality of mine tremors hazard forecasting using neural networks, *Geotech. Geol. Eng.* 24 (5) (2006) 1131–1147.
- [28] J. Kabiesz, B. Sikora, M. Sikora, L. Wróbel, Application of rule-based models for seismic hazard prediction in coal mines, *Acta Montan. Slovaca* 18 (4) (2013) 262–277.
- [29] B. Khaleghi, A.M. Khamis, F. Karray, S.N. Razavi, Multisensor data fusion: a review of the state-of-the-art, *Inf. Fusion* 14 (1) (2013) 28–44.
- [30] M. Kozielski, A meta-learning approach to methane concentration value prediction, in: *Proceedings of the 2016 BDAS*, 2016, pp. 716–726.
- [31] B. Lantz, *Machine Learning with R*, Packt Publishing, 2016.
- [32] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, D. Siegel, Prognostics and health management design for rotary machinery systems – reviews, methodology and applications, *Mech. Syst. Signal Process.* 42 (1) (2014) 314–334.
- [33] C. Li, D. Ai, Automatic crack detection method for loaded coal in vibration failure process, *PLoS One* 12 (10) (2017) e0185750.
- [34] H. Liu, X. Wu, S. Zhang, A new supervised feature selection method for pattern classification, *Comput. Intell.* 30 (2) (2014) 342–361.
- [35] C. Mark, Coal bursts in the deep longwall mines of the United States, *Int. J. Coal Sci. Technol.* 3 (1) (2016) 1–9.
- [36] W. Moczulski, P. Przysała, M. Sikora, R. Zimroz, Modern ICT and mechatronic systems in contemporary mining industry, in: *Proceedings of the 2016 IJCRS*, 2016, pp. 33–42.
- [37] U. Mönks, H. Dörksen, V. Lohweg, M. Hübner, Information fusion of conflicting input data, *Sensors* 16 (11) (2016) E1798.
- [38] Z. Pawlak, A. Skowron, Rudiments of rough sets, *Inf. Sci.* 177 (1) (2007) 3–27.
- [39] K. Pawłowski, K. Kurach, Detecting methane outbreaks from time series data with deep neural networks, in: *Proceedings of the 2015 RSFDGrC*, 2015, pp. 475–484.
- [40] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (8) (2005) 1226–1238.
- [41] D.V. Petrović, M. Tanasijević, V. Milić, N. Lilić, S. Stojadinović, I. Svrkota, Risk assessment model of mining equipment failure based on fuzzy logic, *Expert Syst. Appl.* 41 (18) (2014) 8157–8164.
- [42] R. Polikar, J. De Pasquale, H.S. Mohammed, G. Brown, L.I. Kuncheva, Learn++. MF: a random subspace approach for the missing feature problem, *Pattern Recognit.* 43 (11) (2010) 3817–3832.
- [43] B. Ponte, D. Fuente, J. Parreño, R. Pino, Intelligent decision support system for real-time water demand management, *Int. J. Comput. Intell. Syst.* 9 (1) (2016) 168–183.
- [44] L.S. Riza, A. Janusz, C. Bergmeir, C. Cornelis, F. Herrera, D. Ślęzak, J.M. Benítez, Implementing algorithms of rough set theory and fuzzy rough set theory in the R package 'RoughSets', *Inf. Sci.* 287 (2014) 68–89.
- [45] D. Ruta, L. Cen, Self-organized predictor of methane concentration warnings in coal mines, in: *Proceedings of the 2015 RSFDGrC*, 2015, pp. 485–493.
- [46] J. Rzeszotko, S.H. Nguyen, Machine learning for traffic prediction, *Fundam. Inf.* 119 (3–4) (2012) 407–420.
- [47] M. Sikora, B. Sikora, Improving prediction models applied in systems monitoring natural hazards and machinery, *Appl. Math. Comput. Sci.* 22 (2) (2012) 477–491.
- [48] L. Wróbel, M. Sikora, M. Michalak, Rule quality measures settings in classification, regression and survival rule induction – an empirical approach, *Fundam. Inf.* 149 (4) (2016) 419–449.
- [49] A. Zagorecki, Prediction of methane outbreaks in coal mines from multivariate time series using random forest, in: *Proceedings of the 2015 RSFDGrC*, 2015, pp. 494–500.
- [50] E. Zdravetski, P. Lameski, V. Trajković, A. Kulakov, I. Chorbev, R. Goleva, N. Pombo, N. Garcia, Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering, *IEEE Access* 5 (2017) 5262–5280.
- [51] M. Zhang, A.A. Sawchuk, Motion primitive-based human activity recognition using a bag-of-features approach, in: *Proceedings of the 2012 IHI*, 2012, pp. 631–640.
- [52] Y. Zhang, W. Yang, D. Han, Y.I. Kim, An integrated environment monitoring system for underground coal mines – wireless sensor network subsystem with multi-parameter monitoring, *Sensors* 14 (7) (2014) 13149–13170.