Cluster Discovery from Sensor Data Incorporating Expert Knowledge

Szymon Bobek[1,2][0000−0002−6350−8405], Agnieszka Trzcinkowska[2], Edyta Brzychczy[2], and Grzegorz J. Nalepa[1,2][0000−0002−8182−4225]

[1] Jagiellonian University, Krakow
[2] AGH University of Science and Technology, Krakow
szymon.bobek@uj.edu.pl,
{agnieszka.trzcionkowska,brzych3}@agh.edu.pl, gjn@gjn.re

Abstract. Analysis of sensor data in the industrial setting is commonly performed with the use of data mining methods based on the machine learning algorithms. However, we argue that a proper understanding of this data requires incorporation of expert knowledge. In fact, it is often the case that such an explicit knowledge is available and can be used to enhance the learning process. In this paper we discuss how expert knowledge can be used to validate a machine learning model. More importantly, we demonstrate how a machine learning model can later be used to refine the expert knowledge. We present our framework on a real life use-case scenario from an industrial installation in an underground mine.

Keywords: data mining, clustering

1 Introduction

Industry 4.0 aims at using number of information and communications technology (ICT) solutions for the monitoring and optimization of industrial processes. The installations in modern factories are equipped with a range of sensors gathering data about the operation of the machines involved in these processes. However, an effective analysis of this data can often pose a major challenge. This is where methods of Artificial Intelligence prove to be useful. Moreover, besides the use of data mining techniques in order to build machine learning models, the human expert knowledge regarding the specificity of industrial processes should be used. Our work aims at improving the monitoring the operation of industrial devices. Commonly the monitoring process employs a range of sensors providing monitoring raw data. Currently in many factories this data is only subject to some general statistical summarization and visualization. Our work aims at providing methods for analyzing raw data from sensors monitoring the operation of industrial devices to enable the use of this data for more advanced analysis and ultimately modeling of industrial process on a higher level of abstraction aligned with the background knowledge. In our case, the sensory data obtained from a device can be considered as multidimentional data stream. Our goal is to extract subsequences from this data stream that allow to *identify distinct states of the device* over some periods of time. We aim at utilizing these automatically discovered states to *expand expert knowledge* about the process. We achieve that by combining cluster analysis framework with a symbolic knowledge representation of the device operational states obtained from an expert. As such, we discuss how expert knowledge can be used to validate a machine learning model. Moreover, we demonstrate how a machine learning model can later be used to refine the expert knowledge. While we introduce our approach on a specific use case, we aim at developing methods which are possibly generic. This work is carried out in the CHIST-ERA Pacmel project. [3] The project is oriented at the development of novel methods of knowledge modeling and intelligent data analysis in Industry 4.0. In the project we closely collaborate with several industrial companies providing us with expert knowledge regarding the machinery and industrial installations, as well as large data samples from industrial sensors. In this paper we focus on the case related to the underground mining facilities. Our partner, Famur S.A. [4], is one of the global suppliers of longwall mining machines used in the so-called longwall mining process.

In this work we focus on automated discovery of device states from machinery sensor log to enrich the expert knowledge about machinery operational states. There are several challenges related to the task of automated discovery of device states from data stream. First of all, data in our case is in most cases unlabeled, multidimensional time series varying from 170 to 300 dimensions depending on a vendor and a machinery setup. Features are difficult to interpret, as they include many measurements of operational conditions of a device, such as temperature, currency, oil level, etc. Finally, measurements scales, and types of sensors may vary from device to device (depending on the vendor). Our main goal was to develop a workflow, which would provide a mechanism for detecting device states that can be applied to different types of industrial machinery. We confronted it with states that were discovered with knowledge-based approach to prove its validity and expand the knowledge-base itself. The high-level idea of our approach was depicted in Fig. 1.

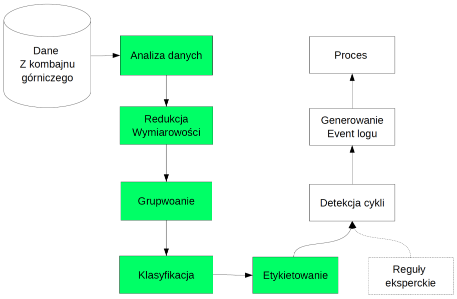


Fig. 1. Workflow of discovery of device states from raw sensor data

[3] See the project webpage at `http://PACMEL.geist.re`.

[4] See the company webpage at `http://www.FAMUR.com.pl`.

In the first step we performed feature engineering to remove data of low quality, and therefore reduce the dimensionality of a problem. After that we applied Principal Component Analysis (PCA) to even further reduce the number of dimensions, by selecting only the features that contributed most to the most relevant components calculated with PCA. This allowed us to use understandable features, while still removing the dimensionality by a reasonable factor. The next step includes additional data enrichment and transformation. Their goal was to include trends and temporal characteristics of the most relevant measurements discovered in previous stage. Finally, we performed clustering of the data obtained from previous steps and evaluated them with labels obtained from expert-based knowledge. These evaluation was combined with an automated enrichment of the knowledge-base, that aimed at splitting or merging states described in the knowledge base with a use of cluster analysis framework.

There is a large number of methods that allow for unsupervised pattern recognition in time series [2]. Most often these methods are concentrated around either time series segmentation [10], point in time clustering [6], or whole time-series clustering [4]. Time series segmentation aims at clustering set of sub-sequences extracted with sliding window, or change-point detection from the original time series. Point in time clustering performs very similar operation, but is not restricted to segments of the original time series, but rather considers each point separately (with some restriction to their temporal dependency). Finally, the whole time-series clustering aims at grouping different time series into clusters of similar characteristics. The time series does not have to be aligned in time, nor be dependent of each other such as in the two former cases.

In this paper we use a special case of point in time clustering. Its main goal is to detect slices of multidimensional time series (subsequences of original time series) that gather points similar with respect to some metric and label these slices with common labels. These labels are later confronted with expert labels and either discarded, or used as a knowledge extensions. In our work we apply term *expert knowledge* to domain knowledge about operational states of machinery. We do not refer this term to expert knowledge in machine learning and data preprocessing. Hence, only extensions are made to the domain knowledge.

Currently, in the context of process-oriented analytic, methods for labeling of raw sensor data are intensively investigated in the process mining domain, as so-called event abstraction [5, 8, 16]. In this scope various approaches can be used: unsupervised learning (e.g., clustering), supervised learning (e.g., classification with labeled data), behavioral patterns analysis and others [5]; see a recent review in [16]. In the mining domain, process analysis based on the raw sensor data is still unrepresented. Initial works related to event abstraction in the mining domain were presented in [3, 14].

The rest of the paper is organized as follows. In Section 2 we provide an introductory description of the use case scenario and describe the expert based approach for labeling the raw sensory data. Section 3 covers data analysis and clustering approaches on data streams. Automated states discovery and knowledge refinement mechanism is presented in section 4. Summary and future works were briefly discussed in Section 5.

2 Expert Knowledge Base

Our use case concerns the operation of a coal mine shearer. A shearer is the main element of longwall equipment and it is used for coal mining and loading on the armored face conveyor (AFC). A shearer consists of two mining heads (cutter drums), placed on the arms, and a machine body containing electric haulage, hydraulic equipment and controls. A shearer is mounted over the AFC. The working shearer moves in two directions: along the longwall face (from the maingate to the tailgate), cutting the coal and due to mining direction – along the length of the longwall panel. There are three main operating states of the shearer: cutting (moving along the longwall face with working drums), moving (moving along the longwall face without working drums) and stoppage. In the case of the considered industrial setting, the knowledge base was encoded as a set of rules obtained from a domain experts, that describe theoretical operation of a coal mine shearer and can be used for the recognition of its activity. The rules describe higher-level operational state of the machine that can be referred to the process of coal extraction presented in Fig. 2.

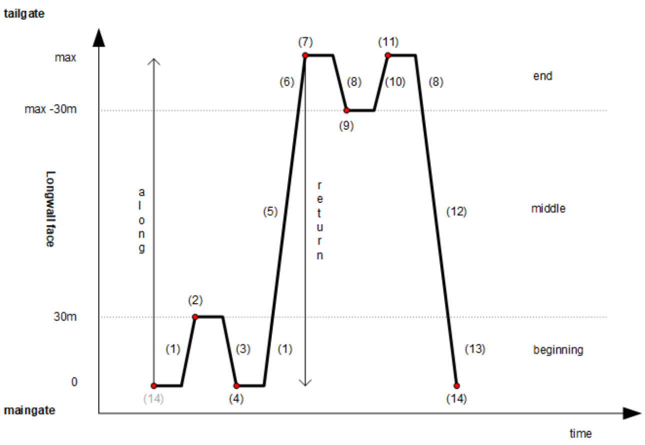


Fig. 2. Model of a shearer cycle

Each activity number given in the Fig. 2 refers to a single identified process stage. Their meaning is as follows:

1. A - cutting into tailgate direction - beginning of the longwall,
2. A - stoppage in ON mode - the beginning of the longwall - (location: 30-40m from the maingate),
3. A - cutting - return to maingate - beginning of the longwall,
4. A - stoppage in ON mode - the beginning of the longwall (location: minimal value - maingate),
5. A - cutting - middle of the longwall,
6. A - cutting into tailgate direction - end of the longwall,
7. A - stoppage in ON mode - end of the longwall (maximal value - tailgate),

8. R - cutting into maingate direction - end of the longwall,
9. R - stoppage in ON mode - end of the longwall (location: 30-40m from the tailgate),
10. R - cutting - return to tailgate - end of the longwall,
11. R - stoppage in ON mode - end of the longwall (maximal value - tailgate),
12. R - cutting - middle of the longwall,
13. R - cutting into maingate direction - the beginning of the longwall,
14. R - stoppage in ON mode - the beginning of the longwall (location: minimal value - maingate).

Each state of the machinery denoted in the picture by an integer value from a range of 1 to 14 can be described by an expert rule encoded in a form of decision tree in Fig. 3.

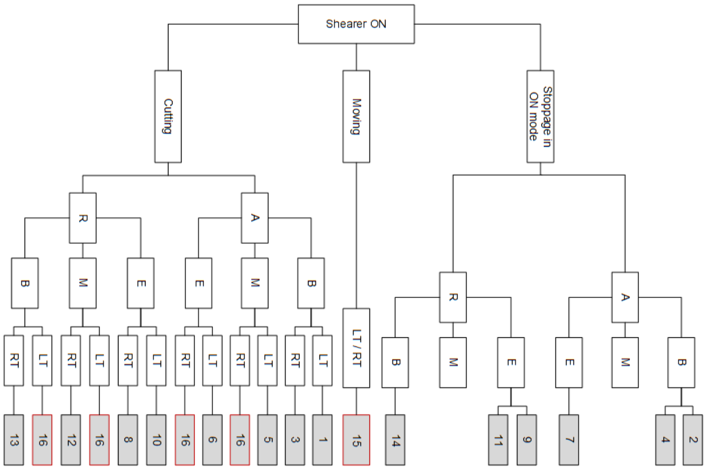


Fig. 3. Rule tree for activity description

The first split in the decision tree corresponds to the state of the shearer. Specific rules denote these states according to haulages and drum currents and shearer speed[5]. The second split is done by part of the cycle: along (A) or return (R). The next split depends on shearer location in the longwall beginning (B), middle (M) or end (E). The last split corresponds to the movement direction of the shearer right (RT) or left (LT). In the case of the definition of *moving* activity - only move direction is taken into consideration. In the case of stoppages, the movement direction split is not applicable. Each branch of the tree can be expanded into rule in a form presented below:

```
IF shearer state = "cutting"
  AND cycle part = "A"
  AND location = "middle"
  AND move direction = "LT"
THEN activity = "5"
```

[5] Due to information policy of collaborating companies, the rules cannot be presented in detail.

The knowledge base contains two more states than in the process from Fig. 2, namely moving without cutting (15) and cutting in opposite direction to the cycle part (16), which are abstract states that does not directly refer to the process of coal extraction but rather machinery movement during that process, yet are need to be modeled.

Rules obtained from the decision tree can be applied directly to label the raw data. Such a labeling can be used in real-life scenario for generating summaries and basic statistics of operation of the sheerer. However, in this approach, some non-typical behavior of the shearer can be lost. This may occur when one of the states (1-16) denoted by the expert rule, encapsulates more specific and highly distinguishable states. These states may correspond for example to abnormal machinery operation due to possible hardware fault or inappropriate device control by its operator. In the next section we present various unsupervised approaches for clusters discovery aiming in improvement of expert rules with potentially new and valuable extensions.

3 Discovering Clusters From Sensor Data

In our case the sensor data is a multidimentional industrial log from the mining machinery, i.e. shearer. It contains 148 features that are raw sensor readings sampled every second. The full length of the data is about one year, however we focused only on one month time span. In the data set both numerical and categorical types of variables exist. Our goal was to automatically distinguish process stages/activities by clustering the raw data into specific clusters that could be base to further analysis of process performance in a form of an event log. In the selection of variables we have applied two approaches: bottom-up (PCA analysis) and top-down (domain experts extensions). In order to do that we first had to cope with low quality of data. This corresponds to *feature engineering* block in Fig. 1. Unfortunately, real industrial data very often are incomplete and noisy. In our case analysed data set 30% of columns are entirely empty, almost 50% of variables have more than 50% of missing values. Only 57% of all variables are suitable for further analysis. We also observed that some of character variables related to security sensors contains only one logical value *False*. This means that such securities did not work even once and these variables were excluded from further analysis.

The rest of the missing values was imputed as follows:

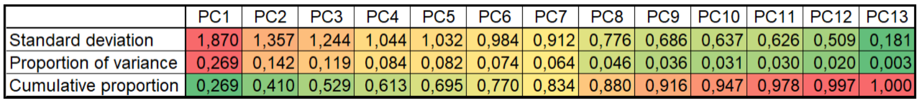
- numerical features with Multivariate Imputation by Chained Equations (MICE) with unconditional mean method [15]. Exception was variable SM-ShearerLocation, in which, due to its specificity, missing values were imputed by interpolation.
- categorical features with mode value.

As outliers identification method we used IQR (inter-quartile range). In case of the shearer location variable all cases below 5 meters were supplemented with correct ones (these anomalies had to be removed due to calibration errors in monitoring the position of the shearer). In the last stage of data cleaning we removed from the further analysis all numerical features with correlation coefficient greater than 0,6.

After data cleaning we performed dimensionality reduction with PCA. This corresponds to the *dimensionality reduction* block in workflow in Fig. 1. Based on identification coverage of variance in dataset by individual principal components as well as analysis of variable loadings (matrix whose columns contain the eigenvectors) the final numeric variables have been selected. For determining the number of principal components was used the cumulative proportion to determine the amount of variance that the principal components explain. The permissible level of cumulative variance is not clearly defined in the literature [1]. There are several criteria used in practice, but there is no objective criterion that clearly indicates which components should be removed. In described approach was used three most common criteria used in practice [7]:

- Cumulative percentage of explained variance of the analyzed variables. We choose the smallest number of principal components for which the sum of their variances constitutes a certain part of the variance of all variables subjected to reduction.
- Kaiser criterion - main components are left that have eigenvalues greater than one (this criterion is used when dealing with a correlation matrix).
- Cattell criterion - based on the Scree plot analysis. This is a graph showing eigenvalues grouped in ascending order.

Based on the analysis of the cumulative percentage of explained variance of the analyzed variables, the highest acceptable percentage, i.e. 91%, i.e. the main components from PC1 to PC9, were selected first. Then, based on the Kaiser Criterion, another main component PC10 was deprived. Scree plot analysis finally confirms the selection of main components from PC1 to PC10 (Fig. 4).



	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Standard deviation	1,870	1,357	1,244	1,044	1,032	0,984	0,912	0,776	0,686	0,637	0,626	0,509	0,181
Proportion of variance	0,269	0,142	0,119	0,084	0,082	0,074	0,064	0,046	0,036	0,031	0,030	0,020	0,003
Cumulative proportion	0,269	0,410	0,529	0,613	0,695	0,770	0,834	0,880	0,916	0,947	0,978	0,997	1,000

Fig. 4. Results of PCA analysis

In order to more accurately interpret the main components, the size and direction of the coefficients for the original variables were analyzed. The higher the absolute value of the coefficient, the more important the corresponding variable is when calculating the component. The value of 0.5 was selected as the absolute value of the coefficient. As a result of PCA analysis (bottom-up approach), we create the following list of numerical variables: SM-ShearerLocation, SM-ShearerSpeed, SM-DailyRouteOfTheShearer, and RP-AverageThree-phaseCurrent. However, we extended this list with variables pointed by domain experts as crucial for activity definition (top-down approach), namely: SM-TotalRoute, LCD-AverageThree-phaseCurrent, RCD-AverageThree-phaseCurrent, LHD-EngineCurrent, RHD-EngineCurrent, and LP-AverageThree-phaseCurrent.

Shortcut at the beginning of variable's name denotes part of the shearer from which data come from (SM - the main body of the shearer, LCD or RCD - respectively left or right cutter drum, LHD or RHD - respectively left or right haulage, LP or RP - respectively left or right pump). We also added artificial features that allowed us to represent context of instances in time. This corresponds to the *feature enhancements* block in workflow presented in Fig. 1. Such features included mean and standard deviation with 3 minutes sliding windows. In the case of categorical features (mainly of boolean type), after data cleaning stage we could take only into consideration two variables: SM-MoveLeft and SM-MoveRight. The final data set contained 12 original variables (10 numerical and 2 categorical) and 27 artificial features. The final step, before evaluation was *clustering* as presented in Fig. 1. We use several different approaches to cluster data, i.e. clustering of: 1) raw numerical variables (RNV), 2) artificial numerical variables (ANV), 3) categorical variables (transformation of numerical variables into categorical) (CV), and 4) mixed variables (MV). In the cluster analysis of numerical and artificial numerical variables we used the K-means algorithm and hierarchical clustering. We examined the range of K from 5 to 15. In the clustering of categorical variables, firstly, we perform the discretization of numerical variables according to specified guidelines and combine them with original categorical variables. The discretization of variables related to the electrical current (regarding to shearer drums and haulages) was carried out according to the following guidelines (based on an expert knowledge and nominal value given in a machinery documentation): 1) Idle value (0-10)% of nominal value, 2) Low load (10-40)% of nominal value, 3) Medium load (40-80)% of nominal value, 4) High load (80-100)% of nominal value, 5) Overload (above 100)% of nominal value. Other numerical variables were discretized to equal five intervals. In clustering, we used *ward.D2* agglomeration method and similarity matrix was created with the *Gower* distance. In the clustering of mixed variables we used K-means algorithm with distinction of movement direction (left-right) and hierarchical approach. Table 1 shows the results of clustering approaches and K value with Silhouette score for each. Visualization of defined clusters on the shearer cycle for the best cluster number for each approach is presented in Fig. 5.

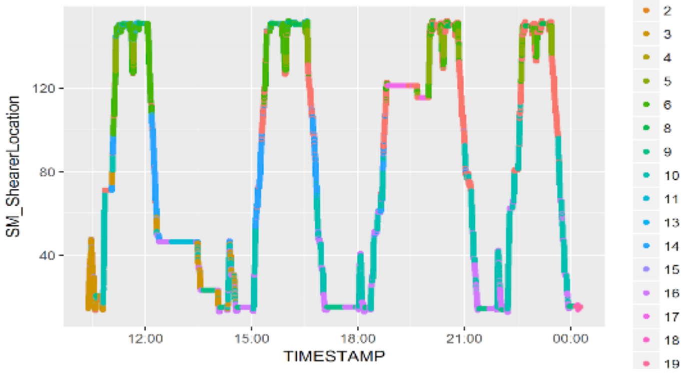


Fig. 5. Optimal clusters in RNV approach

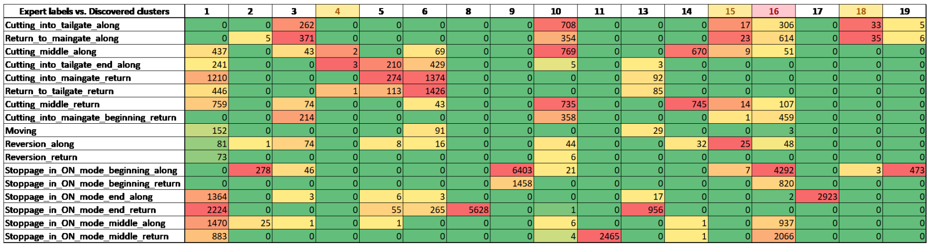
Our calculations bring ambiguous results. Various approaches have a different in Silhouette score. However, the differences (except CV approach) are not so significant. We can observe good homogeneity in low number of clusters (5-6) or higher number (19) of clusters. Relatively low value of Silhouette score for CV approach results from other distance measure (*Gower*). In the next section, we present a comparison of the unsupervised activity (cluster) discovery with knowledge-based activity labeling of the shearer operation process to obtain possible insights for extension of ground expert rules.

4 Evaluation and Domain Knowledge Extension

We used expert labeling for evaluation of automated clustering, as we assume overall correctness of an expert knowledge in this approach. Issues related to handling clustering labeling that is contradictory to expert knowledge labeling is out of the scope of this paper. Our goal was only to refine the knowledge base by splitting and merging some of the states defined by the expert. Therefore we wanted to achieve best possible alignment of the automated clustering with expert labeling and analyze the differences between both to extend the knowledge base. This the last part of the methodology depicted in Fig. 1. We based our evaluation on V-measure [13] enabling conditional entropy-based cluster evaluation in terms of homogeneity and completeness criteria. Table 2 shows the results of V-measure comparison between different clustering approaches and expert labeling on data sample that includes 50000 observations. In terms of V-measures the best clustering results brings the RNV clustering approach with 19 clusters (see Fig. 5). Presentation of cluster distributions versus expert labels is presented in Tab. 4. In each row one can see how a specific expert cluster is divided into more specific clusters by the clustering algorithm. In each column one can observe how multiple expert clusters are combined into one cluster by the machine learning algorithm.

Clustering method	**Experts Labelling - V measure**	**Experts Labelling - homogeneity**	**Experts Labelling - completeness**
RNV	**0.55**	0.56	**0.54**
ANV-AVG	0.47	**0.58**	0,40
ANV-SD	0.18	0.26	0.14
CV	0.43	0.53	0.30
MV	0.42	0.54	0.35

Table 1. V-measure comparison between clustering approaches and expert labeling



Expert labels vs. Discovered clusters	1	2	3	4	5	6	8	9	10	11	13	14	15	16	17	18	19
Cutting_into_tailgate_along	0	0	262	0	0	0	0	0	708	0	0	0	17	306	0	33	5
Return_to_maingate_along	0	5	371	0	0	0	0	0	354	0	0	0	23	614	0	35	6
Cutting_middle_along	437	0	43	2	0	69	0	0	769	0	0	670	9	51	0	0	0
Cutting_into_tailgate_end_along	241	0	0	3	210	429	0	0	5	0	3	0	0	0	0	0	0
Cutting_into_maingate_return	1210	0	0	0	274	1374	0	0	0	0	92	0	0	0	0	0	0
Return_to_tailgate_return	446	0	0	1	113	1426	0	0	0	0	85	0	0	0	0	0	0
Cutting_middle_return	759	0	74	0	0	43	0	0	735	0	0	745	14	107	0	0	0
Cutting_into_maingate_beginning_return	0	0	214	0	0	0	0	0	358	0	0	0	1	459	0	0	0
Moving	152	0	0	0	0	91	0	0	0	0	29	0	0	3	0	0	0
Reversion_along	81	1	74	0	8	16	0	0	44	0	0	32	25	48	0	0	0
Reversion_return	73	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0
Stoppage_in_ON_mode_beginning_along	0	278	46	0	0	0	0	6403	21	0	0	0	7	4292	0	3	473
Stoppage_in_ON_mode_beginning_return	0	0	0	0	0	0	0	1458	0	0	0	0	0	820	0	0	0
Stoppage_in_ON_mode_end_along	1364	0	3	0	6	3	0	0	0	0	17	0	0	2	2923	0	0
Stoppage_in_ON_mode_end_return	2224	0	0	0	55	265	5628	0	1	0	956	0	0	0	0	0	0
Stoppage_in_ON_mode_middle_along	1470	25	1	0	1	0	0	0	6	0	0	1	0	937	0	0	0
Stoppage_in_ON_mode_middle_return	883	0	0	0	0	0	0	0	4	2465	0	1	0	2066	0	0	0

Fig. 6. Expert labels distribution in discovered clusters - RNV approach

Due to the high variability of raw data (data is sampled every one second), during interpretation of created matrix we took into consideration only cells with number of observations above 100. This resulted in excluding clusters: 4 (7 observations in total), 15 (96 observation in total) and 18 (71 observations in total). For the remaining clusters we draw the following conclusions:

- An unequivocal or almost unequivocal matching between cluster and expert label exists in six clusters: 2,8,9,11,17,19.
- There are six clusters including several expert labels, but close in nature and/or place of activity to each other: 3,5,6,10,13,14.
- The most complex situation is in clusters 1 and 16, where cutting and stoppage expert labels are mixed. The analysis of descriptive statistics of these clusters provided to observation that mean values of all variables are comparable, except variable SM-ShearerLocation (for cluster 1 is equal to 104m and for cluster 16 is equal to 25m). Mean values of variable SM-ShearerSpeed are equals to 0.6 [m/s], so we observe in data mostly stoppages events. Existence of other labels probably results from shearer braking near to beginning and end of the longwall face.
- Two clusters were not present in the data sample: 7 and 12.

One can see that there are several expert labels with various clusters matched. It means that in the expert label more specific states can be found (than expert expressed it literally). Thus these findings can be investigated in terms of tree rule extensions (Fig. 3) with an assumption of minimal confidence ratio threshold. We automated that process by providing two algorithms for *split* and *merge* operations. In order to decide on *split* we took the distribution matrix and calculated homogeneity of each cluster with respect to expert labels. The goal was to discard clusters that are large in volume (lots of points covered), but not precise in their fit to specific expert clusters (e.g. cluster 1). We then mean-normalized the rows of the distribution matrix to obtain confidence factors of each potential split. We divided each confidence factor by the homogeneity measure calculated previously to enforce splits that are well fitted to the expert cluster (e.g. cluster 9, 19, 2). As a result we got the *split matrix*. Based on that we selected best splits and their confidence by average confidence of each split. The choice of splits was parameterized by the threshold value. Below the split suggestions were presented for a threshold value 0.3.

```
Cutting_middle_along SPLIT TO [(10, 14), 0.38]
Cutting_into_maingate_return SPLIT TO [(1, 6), 0.42]
Cutting_into_maingate_beginning_return SPLIT TO [(10, 16), 0.40]
Moving SPLIT TO [(1, 6), 0.41]
Stoppage_in_ON_mode_middle_along SPLIT TO [(1, 16), 0.49]
```

In order to decide which clusters should be *merged* we used the $l2$ normalized *split matrix* and calculated cosine similarity between rows in the matrix. As a result we obtained a distance matrix that present high cosine similarity between expert labels that were similarly splited by the *split matrix*. Cosine similarity allowed us to bound the similarity between 0 and 1, and allow for better comparison of cluster matches that differs in number of points (magnitude of the vector). The choice of expert labels to merge was parameterized by the threshold value. Below the merge suggestions were presented for a threshold value 0.8.

```
Return_to_maingate_along
```

```
    MERGE WITH Stoppage_in_ON_mode_middle_return (0.81)
Cutting_into_tailgate_end_along
    MERGE WITH Cutting_into_maingate_return (0.89)
```

The aforementioned split and merge recommendation needs to be revised by an expert and incorporated into the knowledge base manually, as the conditions defining splits and merges are not yet generated automatically. This will allow for the refinement of the original rule tree capturing the expert knowledge. At the time of the writing of the paper, we are still waiting for the response from the company experts. These enhancements are parts of the future plans on extending the framework.

5 Summary and Future Works

In this paper we presented a framework for expert knowledge extension with a usage of clustering algorithms for multidimensional time series. We described how automated mechanism for labeling deceive operational states can be used to refine expert-based labeling. These refinements was defined by us as *splits* and *merges* of expert labeling. We demonstrated the framework functionality on a real use-case scenario that was delivered to us by project partner the Famur S.A. company. Such refined knowledge can further be used to for generating more detailed reports on machine operational states, as well as for detecting abnormal behaviour of the machinery, which was not detected by original expert-knowledge rules. In the further steps, the discovered clusters can be described in detail with the use of descriptive analytic and as result, incorporated in the existing rule tree. This is the main focus of the future works. We plan to exploit the capabilities of state-of-the-art frameworks for explainable AI, such as LIME [11], SHAP [9] and in particular ANCHOR [12], to not only present suggestions on splits and merges, but also generate rule-based explanations on these suggestions that can be easily incorporated into the existing knowledge base in semi-automatic way.

Acknowledgements

The authors would like to thank Dr Jacek Korski from Famur S.A. for his remarks and comments on the paper. The paper is funded from the PACMEL project, the National Science Centre, Poland under CHIST-ERA programme (NCN 2018/27/Z/ST6/03392).

References

1. H. Abdi and L. J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010.
2. S. Aghabozorgi, A. S. Shirkhorshidi], and T. Y. Wah]. Time-series clustering – a decade review. *Information Systems*, 53:16 – 38, 2015.
3. E. Brzychczy and A. Trzcionkowska. Process-oriented approach for analysis of sensor data from longwall monitoring system. In *International Conference on Intelligent Systems in Production Engineering and Maintenance*, pages 611–621. Springer, 2018.

4. A. Dempster, F. Petitjean, and G. I. Webb. Rocket: Exceptionally fast and accurate time series classification using random convolutional kernels, 2019.
5. K. Diba, K. Batoulis, M. Weidlich, and M. Weske. Extraction, correlation, and abstraction of event data for process mining. *WIREs Data Mining and Knowledge Discovery*, 10(3):e1346, 2020.
6. V. Hautamaki, P. Nykanen, and P. Franti. Time-series clustering by approximate prototypes. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008.
7. F. Husson, S. Le, and J. Pagès. *Exploratory Multivariate Analysis by Example Using R*. Chapman & Hall/CRC Computer Science & Data Analysis. CRC Press, 2017.
8. A. Koschmider, D. Janssen, and F. Mannhardt. Framework for process discovery from sensor data. In *EMISA Workshop 2020 Proceedings*.
9. S. M. Lundberg, G. G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S. Lee. Explainable AI for trees: From local explanations to global understanding. *CoRR*, abs/1905.04610, 2019.
10. N. Madicar, H. Sivaraks, S. Rodpongpun, and C. A. Ratanamahatana. Parameter-free subsequences time series clustering with various-width clusters. In *2013 5th International Conference on Knowledge and Smart Technology (KST)*, pages 150–155, 2013.
11. M. T. Ribeiro, S. Singh, and C. Guestrin. “why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
12. M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.
13. A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
14. A. Trzcionkowska and E. Brzychczy. Practical aspects of event logs creation for industrial process modelling. *Multidisciplinary Aspects of Production Engineering*, 1(1):77–83, 2018.
15. S. van Buuren and K. Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software, Articles*, 45(3):1–67, 2011.
16. S. J. van Zelst, F. Mannhardt, M. de Leoni, and A. Koschmider. Event abstraction in process mining: literature review and taxonomy. *Granular Computing*, 2020.