DOI: 10.15439/2015F425
ACSIS, Vol. 5

Window-Based Feature Extraction Framework for Multi-Sensor Data: A Posture Recognition Case Study

Marek Grzegorowski
Faculty of Mathematics,
Informatics and Mechanics,
University of Warsaw,
Banacha 2, 02-097 Warsaw, Poland
Email: M.Grzegorowski@mimuw.edu.pl

Sebastian Stawicki
Faculty of Mathematics,
Informatics and Mechanics,
University of Warsaw,
Banacha 2, 02-097 Warsaw, Poland
Email: Stawicki@mimuw.edu.pl

***Abstract*—The article introduces a novel mechanism for automatic extraction of features from streams of numerical data. It was originally designed for the purpose of processing multiple streams of readings generated by sensors in coal mines. The original research was conducted on methane concentration analysis in the DISESOR project. The article demonstrates an application of the elaborated mechanism for the case of tagging short series of readings from sensors that monitor activities and movements of firefighters during the action with labels corresponding to firefighter activities. The purpose of the experiment was to assess how the automatic feature extraction and construction of classifiers (without parameters tuning and without the use of classifier ensembles) can cope with the competition's task in comparison to other participants.**

I. Introduction

Every day, the surrounding world is being monitored by a still increasing number of sensors. Starting with sensors from our neighborhood as: mobile phones, intelligent home appliances, GPS, automotive sensors, cardio-in watches etc. ending with specialized sensors that support the manufacturing processes deployed in factories, mines or platforms. The velocity of data acquisition makes that the methods of analysis are expected to adapt rapidly to the changes and the emergence of data. On the other hand, the similarity of the nature of the data generated by the sensors appears to allow the construction of generic, reusable mechanisms for data processing and analysis.

The recent emergence of data storage technologies like columnar databases with high level of compression as Infobright [24] and the solutions that can scale up to thousands of machines like MapReduce [8] allow us to store machine generated data that is extremely large. What has to be done at this point, is to develop a generic approach to process data and to introduce a mechanism for automatic (or semi-automatic) knowledge discovery from acquired data in order to support analysts. This aims to reduce the time needed to perform the laborious, manual data analysis.

This article introduces a novel mechanism for automatic extraction of features from streams of numerical data and verifies its effectiveness based on data mining competition results. The elaborated mechanism was originally prepared for the purpose of processing multiple streams of readings generated by sensors in coal mines. The article demonstrates an application of the developed mechanism for the case of the AAIA'15 Data Mining Competition[1]: Tagging Firefighter Activities at a Fire Scene[16] which was the continuation of the previous contest investigating key risk factors for Polish Fire Service [11]. The competition was concerned the process of automatic labels (activities) assignment to a short series of readings from sensors that monitor activities and movements of firefighters during the action. The aim of the competition was to maximize balanced accuracy measure which is defined as an average accuracy within all decision classes while the aim of our research was to assess how the automatic feature extraction and classifiers learning (without parameters tuning) can cope with the competition's task.

Another of our objectives was a requirement that the total effort spent on the data preparation and experiments should be limited, which enables easier management of human resources. The overall time was limited in advance by 2MD (two man days - that is 16 h) which has been recognized as sufficient for researchers to become familiar with the task and to adjust original data representation to a format accepted by the evaluated feature extraction mechanisms. A part of the available time was used for a classifier selection and learning process and was conducted by means of the algorithms available in packages for R programming language[2].

This paper is organised as follows. In Section II the original data set and features extraction mechanisms are presented. In Section III, the assumptions of experiments, an approach to the features selection, the final solution, as well as verified (but finally discarded) approaches to data analysis are shown. In Section IV, the original application of elaborated mechanisms for the extraction of features from multiple streams within the DISESOR project is described. Finally, in Section V a

[1] https://knowledgepit.fedcsis.org/contest/view.php?id=106

[2] See. http://www.r-project.org/

978-83-60810-66-8/$25.00©2015, IEEE

summary of research, conclusions and plans for the nearest future are presented.

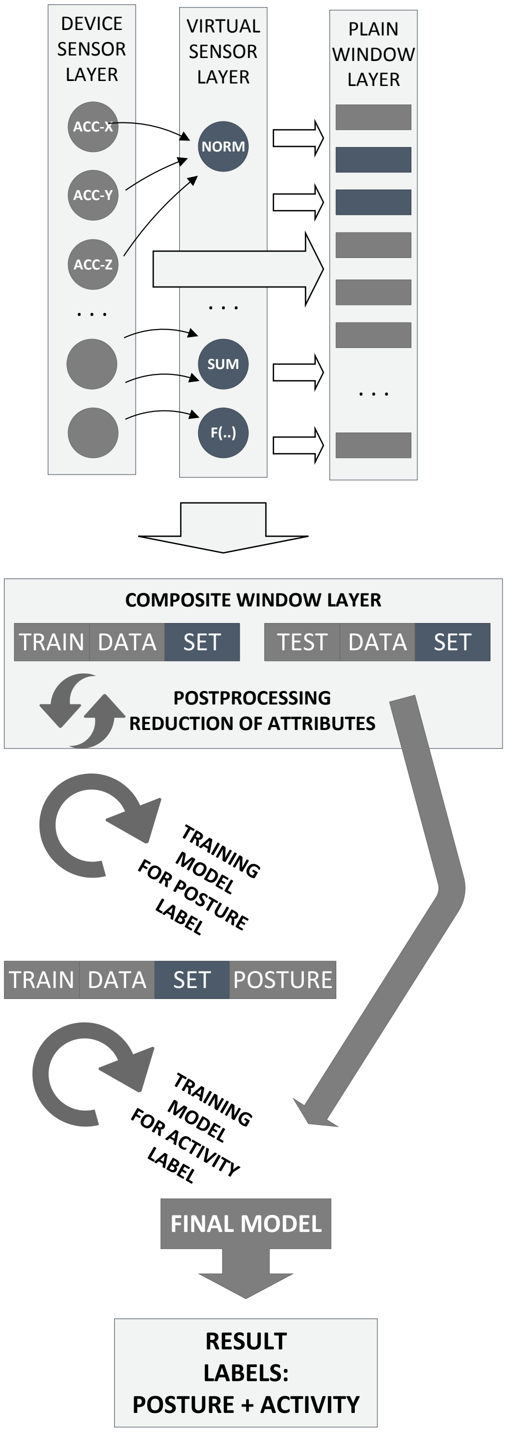

Figure 1. The diagram shows the whole process of feature extraction and model training which was carried out in order to solve the problem of labeling sensor time series with posture and main activity of a firefighter. Model responsible for recognizing a firefighter's posture uses windows constructed on the raw-sensory and virtual-sensory data. The model labeling main activity takes into account both sensory data and a posture label.

II. Data preprocessing

A. Original data set

The data provided in the competition were obtained during training exercises conducted by a group of eight firefighters from the Main School of Fire Service. The sensors placed on a chest were registering vital functions, while the sensors placed on torso, hands, arms and legs were registering movements of a firefighter. Along with recording the data from sensors, all training sessions were also filmed. The video recordings firstly synchronized with the sensor readings, were presented to experts who manually labeled them with actions performed during the exercises. The data were provided as CSV files.

The training and test data sets contain 20,000 rows and 17,242 columns each. A given row in a file corresponds to a short time series with length equal to approximately 1.8 s. The first 42 columns contain basic statistics (aggregations like mean, standard deviation, maximum, minimum, etc.) of data from sensors monitoring a firefighter's vital functions over the given fixed time period. The raw readings for the vital functions were recorded using Equivital Single Subject Kit (EQ-02-KIT-SU-4) fitted with two medical-quality ECG units, heart rate and breath rate units, and thermometers for measuring skin temperature. The remaining columns contain readings from a set of kinetic sensors attached to seven places on a body (torso, hands, both arms and both legs) identified as important during the realization of the main ICRA project's objectives. They are divided into 400 chunks that represent consecutive points in time. Each set is composed of readings from an accelerometer (dynamic bandwith: +/- 16G) and a gyroscope (scale up to 2,000 deg/s), therefore a total number of kinetic sensors are equal to 14. Each sensor of the both types (an accelerometer or a gyroscope) produces three readings x, y, z corresponding to the tree dimensions, hence we have the total number of reading streams equals to 42. A single chunk of columns, therefore, consists of 43 numeric values, from which the first one is time from the beginning of the series and the following 42 values represent the readings from the accelerometers (measured in m/s^2) and gyroscopes (measured in deg/s). An average time difference between consecutive sensory readings in the data is 4.5 ms. The task is even more challenging since the training and test data sets consist of recordings from disjoint groups of firefighters.

The above description shows the details of the values arrangement in the provided data. We considered each row as a separate data set containing readings from many sensors. As described above, values from the vital sensors were aggregated externally, but the kinetic ones are provided in the raw form of time series. Let us present a fragment of an example in a visual form of data plots to better illustrate the amount of available data and their internal dependence. The references to the sensor readings are consistent with the naming from metadata provided by the organizers. There are seven places on a body that the sensors were placed on, i.e. left leg, right leg, left hand, right hand, left arm, right arm, and torso. The body areas corresponds to the following name prefixes:

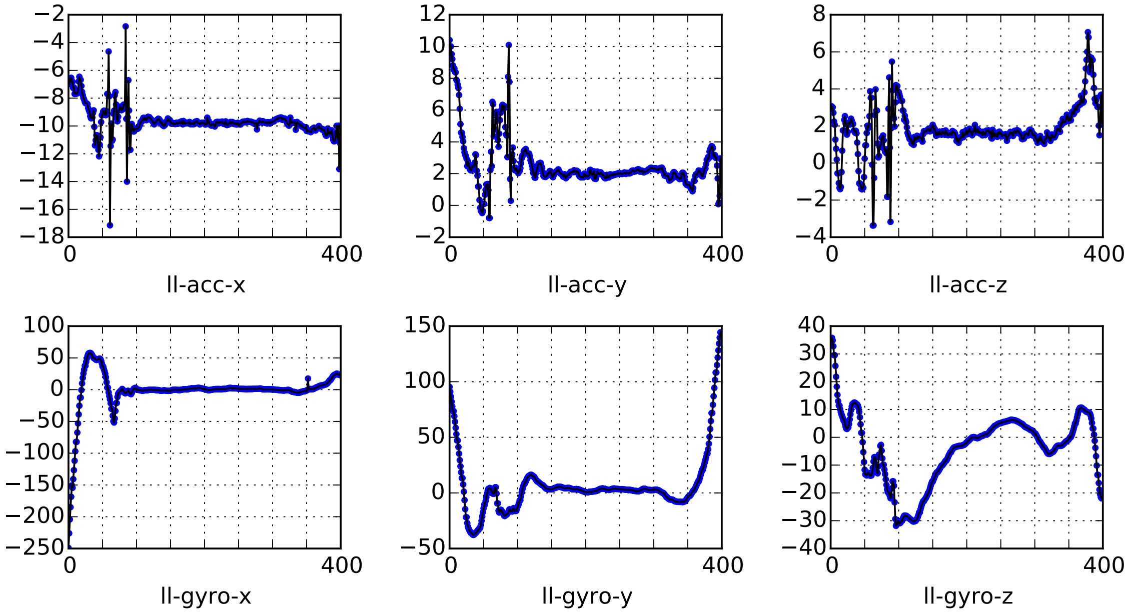

Figure 2. A fragment of an example row

$ll, rl, lh, rh, la, ra, torso$. An name infix *acc* or *gyro* refers to an accelerometer or gyroscope type of sensors. Finally, a suffix x, y, or z names the axis from which the readings came from. In Figure 2 we present an example row that was tagged in the data with "standing" and "no_action" labels describing a posture of a firefighter and his current activity. The figure contain six time series (each consisted of 400 values that correspond to approximately 1.8 s) from the set of sensors placed on a left hand of a subject performing an exercise.

B. Feature extraction

In the process of development of our feature extraction system we decided to follow the sliding window method. In general, for a given set of readings we put a window of a fixed *length* – the size of a window, i.e., a number of readings or a time interval, which travels through the values from the beginning to the end. We can control the amount of processed windows not only by setting the window length but also defining the *offset* for the consecutive windows – the extent to which the consecutive windows overlap to each other. Figure 3 presents four examples of sliding window set-ups. The first example, marked in red, shows the situation when the length of a sliding window is equal to the offset. The green and blue examples show the consecutive positions of a sliding window when the offset is equal to $\frac{1}{2}$ and $\frac{1}{3}$, respectively, of the length. The system is also capable to express the situation when the offset is greater than the length – the example marked in cyan.

For each basic window that is created during the process of moving a sliding window through the time series a defined aggregate function is applied. This step of the process may be adjusted for the actual task by supplying a specific implementation. The following list presents features which are calculated to represent the time series in a window:

- fill – a ratio of correct readings in the window = $\frac{nValid}{n}$,
- firstValue – a value of the first reading in the window,
- lastValue – a value of the last reading,
- max – a maximum value of the readings in the window,
- maxMinDiff – a difference between the max and min,
- mean – a mean value of readings in the window,
- min – a minimum value of the readings in the window,
- n – a total number of readings in the window,
- nValid – a number of valid readings in the window
- percentile25 – a percentile 25% for the readings,
- percentile5 – a percentile 5% for the readings,
- percentile50 – a percentile 50% for the readings (median),
- percentile75 – a percentile 75% for the readings,
- percentile95 – a percentile 95% for the readings,
- percentiles5Diff – a subtraction of the percentiles 95% and 5%,
- sourceFullId – a data source identifier included in the statistics of the window, e.g. ID or a name of the sensor,
- stdDev – a standard deviation of the readings,
- windowEndDate – a window end date
- windowEndMillis – an end timestamp of the window,
- windowMetaInfo – a meta information of the sliding window configuration, encoded in a form of a string, e.g. "o60l60" is equivalent to $offset = 60$ and $length = 60$,
- windowStartDate – a window start date,
- windowStartMillis – a start timestamp of the window,

A sliding window in a fixed position for which the aggregate function was applied and produced the statistics is referred later as a basic (or plain) window. An example of a basic window for an axis x of the sensor placed on a left leg of a firefighter is presented in Table I.

For the purpose of the competition we have processed the data with three layouts of a sliding window. An illustration of our choice is presented in Figure 4. We have decided to

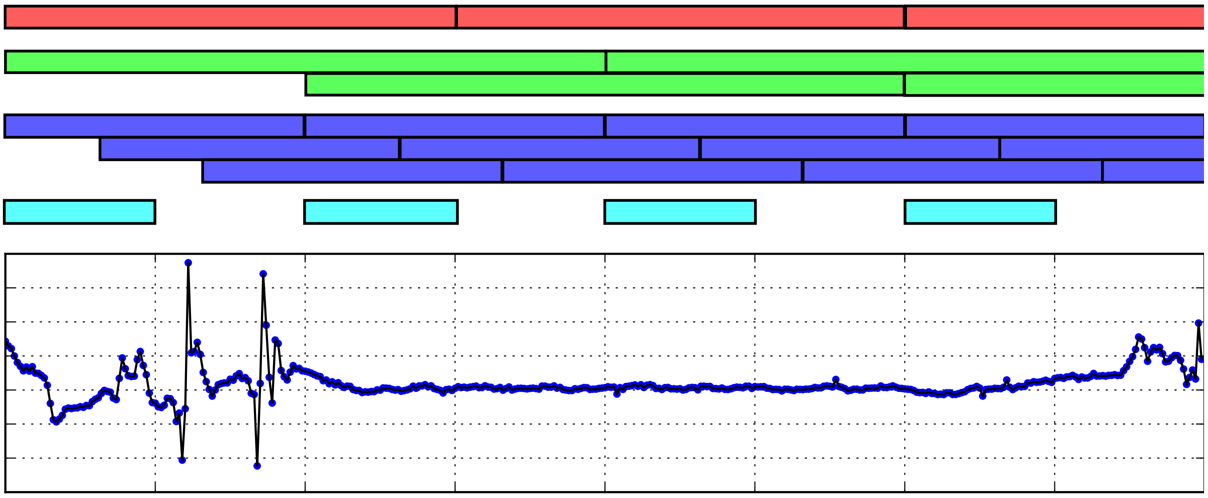

Figure 3. A set of examples showing the possible set-ups. Sliding windows are defined by a *length* and an *offset*. The length determines the size of a window, whether it is a fixed number of readings contained in a window or a fixed time interval. The offset is the extent to which the consecutive windows overlap to each other. The example marked in red shows the situation when the length of a sliding window is equal to the offset. The green and blue examples show the consecutive positions of a sliding window when the offset is equal to $\frac{1}{2}$ and $\frac{1}{3}$ of the length. The example marked in cyan illustrates the situation when the offset is twice as large as the length (or in general just greater) of a sliding window.

calculate statistics for each row by splitting each time series to 1, 2 or 5 consecutive non-overlapping windows.

We have described earlier in the section the capabilities of the feature extraction system to express different layouts of a sliding window in terms of its length and offset. If there is more than one window generated for the time series we can extract additional features in addition to those included in a basic window statistics. We have implemented also inter-window stats extraction, i.e., a set of values that express the changes between a pair of consecutive windows. We have introduced the following inter-window stats:

- firstFill – a ratio of correct readings in the first window,
- firstN – a total number of readings in the first window,
- firstNValid – a number of valid readings in the first window,
- firstWindowDate – a start date in the first window,
- firstWindowMillis – a start timestamp in the first window,
- maxDiff – a difference between *max* statistics in the windows,
- meanDiff – a difference between *mean* statistics in the windows,
- minDiff – a difference between *min* statistics in the windows,
- percentile25Diff – a difference between *percentile25* statistics in the windows,
- percentile5Diff – a difference between *percentile5* statistics in the windows,
- percentile50Diff – a difference between *percentile50* statistics in the windows,
- percentile75Diff – a difference between *percentile75* statistics in the windows,
- percentile95Diff – a difference between *percentile95* statistics in the windows,
- secondFill – a ratio of correct readings in the first window,
- secondN – a total number of readings in the second window,
- secondNValid – a no. of valid readings in the second window,
- secondWindowDate – a start date in the second window,
- secondWindowMillis – a start timestamp in the second window,
- sourceFullId – a data source identifier,
- windowMetaInfo – a meta information of the sliding window,

	stat	value
1	fill	1
2	firstValue	-7
3	lastValue	-11.2
4	max	-2.8
5	maxMinDiff	14.3
6	mean	-9.6
7	min	-17.1
8	n	400
9	nValid	400
10	percentile25	-9.9
11	percentile5	-10.8
12	percentile50	-9.8
13	percentile75	-9.5
14	percentile95	-7.6
15	percentiles5Diff	3.2
16	sourceFullId	ll-acc-x
17	stdDev	1.1
18	windowEndDate	2015-05-03 00:06:40
19	windowEndMillis	1430604400000
20	windowMetaInfo	o4001400
21	windowStartDate	2015-05-03 00:00:00
22	windowStartMillis	1430604000000

Table I
AN EXAMPLE OF AGGREGATION FUNCTION COMPUTATION – A BASIC WINDOW STATS FOR THE FIRST ROW OF THE TRAINING DATA.

An example of the inter-window stats for an axis x of the sensor placed on a left leg of a firefighter is presented in Table II. A sliding window configuration used in the example, i.e., the length of the window is equal to its offset, has produced two basic non-overlapping windows that split the time series from a given row into two halves.

C. Virtual sensors

According to the task description, the kinetic sensors (accelerometers and gyroscopes) used during the exercises have symmetric scales with 0 as their neutral reading. The specificity of the firefighter activities like walking, running, moving up the stairs or ladder, may cause the readings to be more

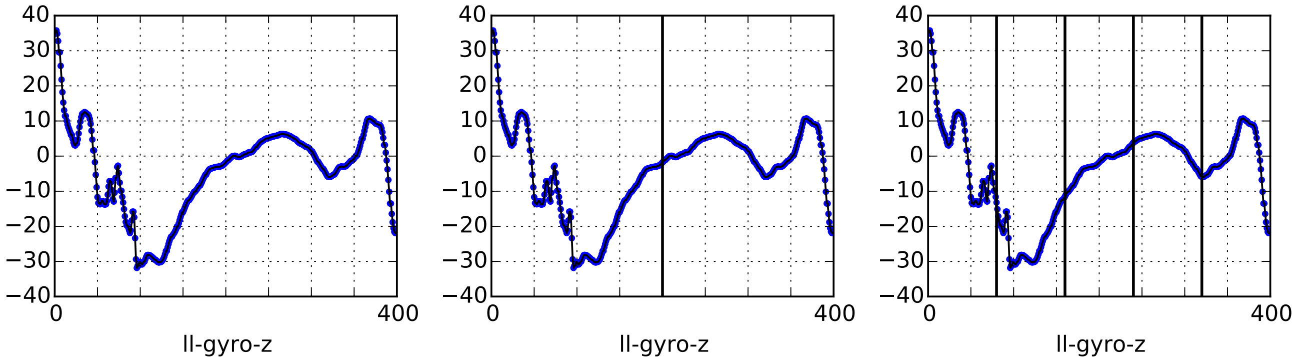

Figure 4. An illustration of the sliding window configurations applied in our solution. We have decided to process the time series with a varied granularity, ranging from the statistics computed for the whole time series, to calculate them for 2 or 5 shorter, non-overlapping windows which divided the time series to the parts of equal length.

	stat	value
1	firstFill	1
2	firstN	200
3	firstNValid	200
4	firstWindowDate	2015-05-03 00:00:00
5	firstWindowMillis	1430604000000
6	maxDiff	6.6
7	meanDiff	0.6
8	minDiff	-4
9	percentile25Diff	0.3
10	percentile50Diff	0.2
11	percentile5Diff	-0.4
12	percentile75Diff	0.8
13	percentile95Diff	2.7
14	secondFill	1
15	secondN	200
16	secondNValid	200
17	secondWindowDate	2015-05-03 00:03:20
18	secondWindowMillis	1430604200000
19	sourceFullId	ll-acc-x
20	windowMetaInfo	o2001200

Table II
An example of inter-window statistics

significant when considered as a group, e.g. a whole tuple (x,y,z) from a given sensor rather than separate readings x, y, and z, to express the intensity of the movement. We have decided to introduce a concept of *virtual sensors*. Besides applying the aggregate functions to the original time series available in the delivered files, we have implemented an idea of creating artificial time series derived from the original ones. The virtual sensors are created on the basis of one or more time series from other sensors (whether original or virtual) after applying a particular function. In our solution, we decided to create virtual sensors for readings from all accelerometers and gyroscopes' axes separately, applying an *abs* (absolute value) function. We created also virtual sensors for readings grouped in tuples (x,y,z) for each kinetic sensor – computing the Manhattan and Euclidean norms for the (x,y,z) vectors. An example that illustrates the concept of virtual sensors that we have used in our solution can be seen in Figure 5.

After all basic windows for original and virtual sensors that comes from a given data row are calculated, they are joined (in the sense of appending all their values) together, forming a row of data that will serve as an input for further steps of data analysis and experiments.

III. Experiments

A. Evaluation

The submitted solutions were evaluated using the balanced accuracy measure which is defined as an average accuracy within all decision classes. It was computed separately for the labels describing the posture and main activities of firefighters. The final score is a weighted average of balanced accuracies computed for those two sets of labels and is defined as follows:

$$score(s) = \frac{BAC_p(s) + 2 \cdot BAC_a(s)}{3}.$$

Where BAC_p is the balanced accuracy for labels describing the posture and BAC_a for the main activity. Precise definition of balanced accuracy is as follows:

$$BAC(preds, labels) = \frac{\sum_{1<i<l} ACC_i(preds, labels)}{l}$$

$$ACC_i(preds, labels) = \frac{\mid j : preds_j = labels_j = i \mid}{\mid j : labels_j = i \mid}.$$

B. Constraints

We considered the competition as a good opportunity to verify the developed mechanisms of automation of knowledge discovery process and their usefulness in the production environment. Therefore, working on the solution we have imposed a few additional constraints and requirements. All have been set arbitrarily for the issue of labeling firefighters activity. We consider them to be satisfactory for the task:

1) Overall working time, to be spent on solving of the problem by all members of the research team must not exceed a total of 2MD.

2) The overall time required to train the classifiers must not exceed the total of 10 minutes. In case of classifiers which are mutually independent this is 10 minutes for training each of them, since the process can be run in parallel.
3) The time required to pre-process a single row of data to a format accepted by classifier and assignment of both labels must not exceed one second.

The first of the imposed restrictions is intended to help to verify whether it is possibile to immediately familiarize analysts with both data and the problems. In the simulated case, two analysts were working on adaptation of the data provided in the new format to the already existing mechanisms. Possibility to adapt quickly to new data and to new expectations while maintaining a satisfactory accuracy of the model is very important especially in the threats monitoring.

The second point poses a constraint on the time that is necessary to re-train the model on the new data, in case after a certain time the quality of the assessment has fallen below the a predetermined score level due to, e.g. concept shift/drift [6]. We assumed that the time required to re-training the model should not exceed 10 minutes. Nevertheless, we consider this point to be the least important and in our opinion exceeding proposed limit should not disqualify the approach. However, in the final embodiment, the total time of training classifiers did not exceed 7 minutes, wherein the classifiers are independent and can be trained simultaneously.

The last point we consider to be the most important because it imposes limits on the permissible delay in operation of preprocessor and classifiers when acting in a production environment. According to the assumptions maximum delay between data collection and complete processing and labeling of single row of data should not exceed one second. This is one of the main reasons for excluding from consideration all object based methods as well as heavy classifier ensembles. Generation of all the features, including those for both: raw and virtual-sensors readings, took approximately 450 milliseconds per a single csv file row. The postprocessing and assignment of the labels has been performed in the R - software environment for statistical computing and consisted of: importing data (overall 30 seconds per 20000 rows of test data set), feature selection (overall 10 seconds per 20000 rows of test set) and labeling (classification with SVM took overall of 70 seconds for both labels for 20000 rows).

C. Post processing

Generated data sets have the following quantity of attributes for each of 20000 objects, depending on configuration:

- 2199 – one sliding window per short time series
- 6315 – two sliding windows per short time series
- 18663 – five sliding windows per short time series

Making a total of 27177 attributes [27] from the conditional- and inter-sliding windows constructed for both raw and virtual sensors. Elements of the automatic feature selection and reducing the number of attributes are in the study phase and still have not been introduced to the data processing mechanisms. Hence, the feature selection was carried out manually.

In a first step, all features exhibiting signs of identifiers and all constants values, that is: fill, n, nValid, sourceFullId, windowEndDate, windowEndMillis, windowMetaInfo, windowStartDate, windowStartMillis have been removed from the prepared data set. We have also removed maximum and minimum of values in windows to limit the influence of outliers on the final result. After applying the model on acquired attributes of training data set we have noticed that the model has been extremely overfitted. As the main reason for this, we find the fact that the training data set was prepared based on the observation of a small number of firefighters, hence data could not contain all possible patterns of motor behavior and vital signs. This observation led us to change our approach and forced to look for features that maintain a quality of prediction for test set.

In the process of feature selection we used a wrapper approach[13]. We have been progressively enlarging the number of utilized features and making periodic evaluations, after each step we have either remained the selected features or we resigned from them, depending on the result of the evaluation. Ultimately, the SVM was run on the 163 attributes for the classifier that labels the objects with posture and with one additional attribute (the computed posture label) for the second SVM model which classifies the data with a main activity. Beyond selected features in sliding windows, described in Section II final set of objects includes additional attributes to exclude the symmetry of right- and left-handed people e.g. the sum of the selected features for the left and right hand as well as sum for the left and right leg. This is very important since the training samples were created basing on the behavior of different people than the test samples. Moreover, training and test set data were acquired during observation of small group of firefighters, hence the training sample could not contain all possible patterns. The situation when training set differs significantly from the test set forced us to make additional step during verification of selected attributes.

D. Classifier training and labeling

Because of the pre-processing of data that has been provided in the competition, the real problem of monitoring the firefighters activities, which is originally associated with processing of streams of sensor readings [10] that constituting time series [2], has been reduced for the problem of classification [25], more precisely to multi-labeling [26]. Original sensor readings has been pre-processed and subdivided into frames [21], [28] of given length and made available in a csv file. To apply the developed feature extraction mechanism each row of the csv file has been split into short time series of readings from sensors respectively to csv header names: "ll-acc-x", "ll-acc-y", "ll-acc-z", etc. and passed as an input stream to the feature extraction mechanism. Eventually, we obtained a set of elaborated features ready for multi-labeling [14].

During data analysis, not only the conditional variables have been inspected but also posture and activity labels. The preliminary conclusions of labels aggregation allowed to state that there is a huge imbalance [29] in classes defined by

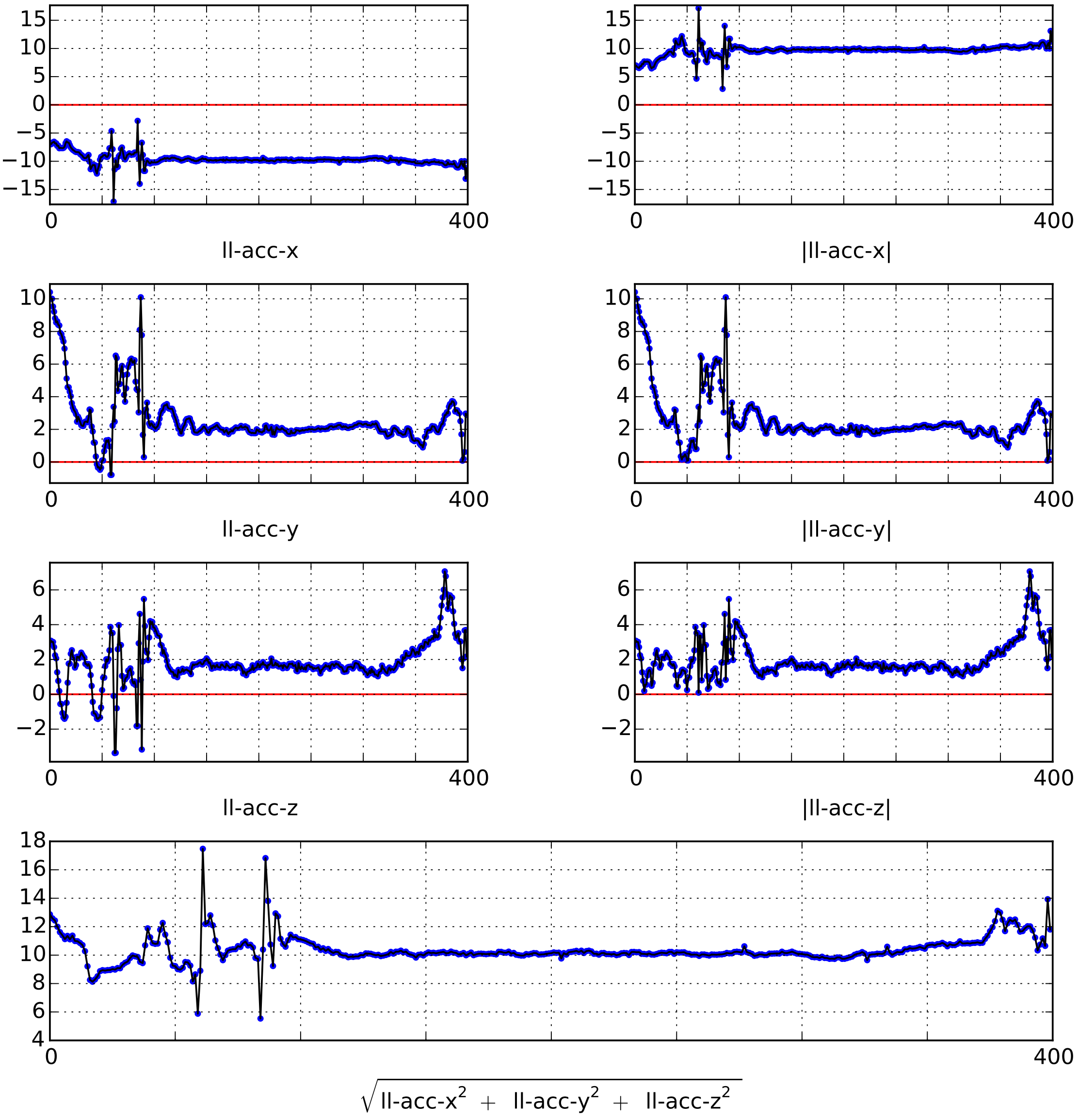

Figure 5. An example of deriving virtual sensors by applying an absolute value function and the Euclidean norm to the original time series.

particular labels and that the labels for firefighters posture and activity are not independent [20] and there is a connection between them [19]. The application of label power-set methods [23], [30] did not provide satisfactory results but classifier chains[22] improved the achieved score significantly. The way in which assessment of solutions was defined, that is uneven importance of labels for posture and activity encouraged to consider various concepts like a multilabel classification with label ranking [9] or a graded multilabel classification [5].

Experiments have been implemented and carried out in the R software environment. We have experimented with the following classification algorithms: rPart (decision trees [4]), rFerns (random forests [3]) and e1071 (support vector machines [1]). The final solution is based on SVM. While learning classifiers we have used the relationship between labels by training two SVM models on slightly different data.

Model 1, which recognizes posture, is SVM with 4356 support vectors. Model has been trained on the basis of the features described above with the default parameters, that is:

- SVM-Type: C-classification
- SVM-Kernel: radial
- Cost: 1
- Gamma: 0.006134969

$$\begin{aligned} &model1 \leftarrow svm(posture \sim ., \\ &\quad data = trainSet[, c(selectedFeatures, posture)]); \end{aligned}$$

Model 2, which recognizes the main activity, is SVM with 5011 support vectors. Model has been trained on data enriched by the posture label with the default parameters, that is:

- SVM-Type: C-classification
- SVM-Kernel: radial
- Cost: 1
- Gamma: 0.005952381

$$\begin{aligned} &model2 \leftarrow svm(activity \sim ., \\ &\quad data = trainSet[, c(selectedFeatures, posture, activity)]) \end{aligned}$$

During labeling, data were firstly described with the posture label, and after that with the main activity label:

$$\begin{aligned} &testLabelsForPosture \leftarrow predict(model1, \\ &\quad newdata = testSet[, selectedFeatures], \\ &\quad type = "class"); \\ &testSet\$posture \leftarrow testLabelsForPosture; \\ &testLabelsForActivity \leftarrow predict(model2, \\ &\quad newdata = testSet[, c(selectedFeatures, posture)], \\ &\quad type = "class"); \end{aligned}$$

IV. DISESOR

The most significant application of the presented solution for automated feature extraction is the ongoing DISESOR project. DISESOR aims to build a decision support system for threats monitoring and early warnings in coal mines.

Nowadays, the coal mining is playing a crucial role on Polish energy market and is employing hundreds of thousands of people. Coal mines are well equipped with monitoring, supervising and dispatch systems connected with machinery, devices and transport facilities. There are a lot of systems that support essentially different aspects of the mine operation, e.g.: ARES, ARAMIS, HESTIA for seismo-acoustic monitoring; RODOS, ALFA for quality control, MAKS, Ergon, Hades for machinery monitoring; SMP, STAR, CTT, UTS, Venturon, Univers for risk control, ZEFIR, THOR, sD2000 - central systems and many, many others. Each of these gathers readings from specific sensors placed in mines, depending on their domain: methane sensor, CO and CO_2 sensor, seismic sensor, shearer state sensors etc. Assembly of a variety of data from multiple systems enables performing a wide-ranging analysis.

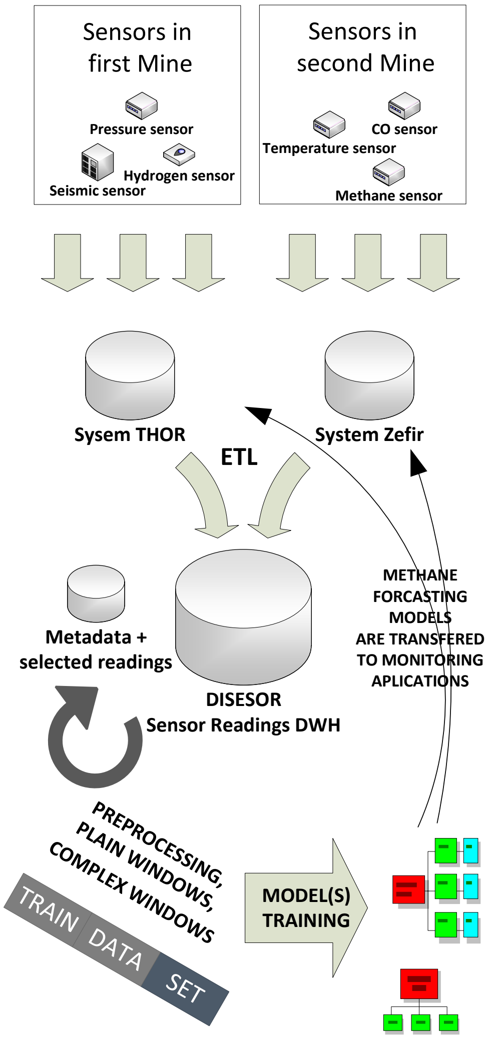

Figure 6. DISESOR ETL process collects sensor readings from mine monitoring systems like THOR or ZEFIR. Raw data is cleand and after preprocesing predictive models are generated.

Monitoring systems are developed by many providers what causes problems with integration and proper interpretation of the data, therefore there is need to deploy a decision support system integrating different aspects of coal mine operations, what is the main task of the DISESOR system. The high level design of DISESOR takes into account the data cleaning process, the process of building data mining models and on-line predictive reasoning for the latest data readings. The most important use cases of the DISESOR system are:

- The assessment of seismic hazard probabilities in the vicinity of the mine.
- Forecasting dangerous increase of the methane concentration in the mine shafts.
- Detection of endogenous fires and conveyor belts fires.
- Detecting anomalies in the consumption of media.
- Diagnostics of machines: roadheaders and shearers.

V. Conclusions and Further Research

The developed feature extraction system can be configured to accept a data set consisted of readings from multiple sensors. The algorithm that builds sliding windows divides reading streams into consecutive fragments and then processes each of them separately. This approach allows for effective parallelization of the whole feature extraction process. However, there are some important issues that have not been addressed in a prepared solution or have been taken into account in a very simplified manner, e.g. a quantization of real value attributes [17], [18] or an attribute selection [7], [12] which we recognize as very important elements of a knowledge discovery [15] process. We are going to extend the discussed mechanisms with modules covering those issues in the nearest future.

The conducted experiments showed that the features prepared by the elaborated mechanism are suitable for machine learning algorithms, which in the next step can give very promising results without neither long lasting manual data cleaning nor classifier tuning. The results of experiments turned out to be significantly better than the baseline solution. Therefore, it seems that the elaborated system is prepared to work in production. However, there is still a lot of space for further improvements since results achieved by other participants in case of manual transformation of data and tuning of classifiers turned out to be even better.

VI. Acknowledgements

This research was partially supported by Polish National Science Centre grant DEC-2012/05/B/ST6/03215 and by Polish National Centre for Research and Development grants PBS2/B9/20/2013 and O ROB/0010/03/001.

References

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[2] G. E. P. Box and G. Jenkins. *Time Series Analysis, Forecasting and Control*. Holden-Day, Incorporated, 1990.

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[5] W. Cheng, K. Dembczynski, and E. Hüllermeier. Graded multilabel classification: The ordinal case. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning, June 21-24, 2010, Haifa, Israel*, pages 223–230. Omnipress, 2010.

[6] J. Coble and D. J. Cook. Real-time learning when concepts shift. In J. N. Etheredge and B. Z. Manaris, editors, *FLAIRS Conference*, pages 192–196. AAAI Press, 2000.

[7] C. Cornelis, R. Jensen, G. H. Martín, and D. Ślęzak. Attribute selection with fuzzy decision reducts. *Inf. Sci.*, 180(2):209–224, 2010.

[8] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, Jan. 2008.

[9] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker. Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153, Nov. 2008.

[10] M. Grzegorowski. Scaling of complex calculations over big data-sets. In D. Ślęzak, G. Schaefer, S. T. Vuong, and Y. Kim, editors, *Active Media Technology - 10th International Conference, AMT 2014, Warsaw, Poland, August 11-14, 2014. Proceedings*, volume 8610 of *Lecture Notes in Computer Science*, pages 73–84. Springer, 2014.

[11] A. Janusz, A. Krasuski, S. Stawicki, M. Rosiak, D. Ślęzak, and H. S. Nguyen. Key risk factors for polish state fire service: a data mining competition at knowledge pit. In M. Ganzha, L. A. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, Warsaw, Poland, September 7-10, 2014.*, pages 345–354, 2014.

[12] A. Janusz and D. Ślęzak. Rough set methods for attribute clustering and selection. *Appl. Artif. Intell.*, 28(3):220–242, Mar. 2014.

[13] A. Janusz and S. Stawicki. Applications of approximate reducts to the feature selection problem. In *Rough Sets and Knowledge Technology - 6th International Conference, RSKT 2011, Banff, Canada, October 9-12, 2011. Proceedings*, pages 45–50, 2011.

[14] W. Jiang, Z. W. Ras, and A. Wieczorkowska. Clustering driven cascade classifiers for multi-indexing of polyphonic music by instruments. In Z. W. Ras and A. Wieczorkowska, editors, *Advances in Music Information Retrieval*, volume 274 of *Studies in Computational Intelligence*, pages 19–38. Springer, 2010.

[15] K. Kreński, A. Krasuski, M. Szczuka, and S. Łazowy. Granular knowledge discovery framework for fire and rescue reporting system. *Intelligent Decision Technologies*, pages 1–12, 2014.

[16] M. Meina, A. Janusz, K. Rykaczewski, D. Ślęzak, B. Celmer, and A. Krasuski. Tagging firefighter activities at the emergency scene: Summary of aaia'15 data mining competition at Knowledge Pit. In M. Ganzha, L. A. Maciaszek, and M. Paprzycki, editors, *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, 2015. In print September 2015.

[17] H. S. Nguyen. On efficient handling of continuous attributes in large data bases. *Fundam. Inf.*, 48(1):61–81, Oct. 2001.

[18] H. S. Nguyen. On exploring soft discretization of continuous attributes. In S. K. Pal, L. Polkowski, and A. Skowron, editors, *Rough-Neural Computing*, Cognitive Technologies, pages 333–350. Springer Berlin Heidelberg, 2004.

[19] S.-H. Park and J. Fürnkranz. Multi-label classification with contraints. In *Proceedings of the workshop on Preference Learning at ECML PKDD'08*, Antwerp, Belgium, 2008.

[20] S.-H. Park and J. Fürnkranz. Multi-Label Classification with Label Constraints. Technical report, Knowledge Engineering Group, TU Darmstadt, 2008.

[21] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping. *ACM Trans. Knowl. Discov. Data*, 7(3):10:1–10:31, Sept. 2013.

[22] J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Mach. Learn.*, 85(3):333–359, Dec. 2011.

[23] J. Read, A. Puurula, and A. Bifet. Multi-label classification with meta-labels. In R. Kumar, H. Toivonen, J. Pei, J. Z. Huang, and X. Wu, editors, *2014 IEEE International Conference on Data Mining, Shenzhen, China, December 14-17, 2014*, pages 941–946. IEEE, 2014.

[24] D. Ślęzak and V. Eastwood. Data warehouse technology by infobright. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 841–846, New York, NY, USA, 2009. ACM.

[25] D. Ślęzak and A. Janusz. Ensembles of bireducts: Towards robust classification and simple representation. In T. Kim, H. Adeli, D. Ślęzak, F. E. Sandnes, X. Song, K. Chung, and K. P. Arnett, editors, *Future Generation Information Technology - Third International Conference, FGIT 2011 in Conjunction with GDC 2011, Jeju Island, Korea, December 8-10, 2011. Proceedings*, volume 7105 of *Lecture Notes in Computer Science*, pages 64–77. Springer, 2011.

[26] D. Ślęzak, A. Janusz, W. Świeboda, H. S. Nguyen, J. G. Bazan, and A. Skowron. Semantic analytics of pubmed content. In *Information Quality in e-Health - 7th Conference of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2011, Graz, Austria, November 25-26, 2011. Proceedings*, pages 63–74, 2011.

[27] M. S. Szczuka and D. Ślęzak. How deep data becomes big data. In *Joint IFSA World Congress and NAFIPS Annual Meeting, IFSA/NAFIPS, Edmonton, Alberta, Canada, June 24-28, 2013*, pages 579–584, 2013.

[28] A. Wieczorkowska, J. Wróblewski, D. Ślęzak, and P. Synak. Problems with automatic classification of musical sounds. In *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'03 Conference held in Zakopane, Poland, June 2-5, 2003*, pages 423–430, 2003.

[29] E. S. Xioufis, M. Spiliopoulou, G. Tsoumakas, and I. Vlahavas. Dealing with concept drift and class imbalance in multi-label stream classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1583–1588. AAAI Press, 2011.

[30] Y. Yang and S. Gopal. Multilabel classification with meta-level features in a learning-to-rank framework. *Machine Learning*, 88(1-2):47–68, 2012.