DOI: 10.26411/83-1734-2015-4-40-10-18

Predictive Business Process Monitoring with Tree-based Classification Algorithms

Tomasz Owczarek, Piotr Janke
Silesian University of Technology, Poland

Predictive business process monitoring is a current research area which purpose is to predict the outcome of a whole process (or an element of a process i.e. a single event or task) based on available data. In the article we explore the possibility of use of the machine learning classification algorithms based on trees (CART, C5.0, random forest and extreme gradient boosting) in order to anticipate the result of a process. We test the application of these algorithms on real world event-log data and compare it with the known approaches. Our results show that.

Keywords: business process, prediction, classification, random forest, gradient boosting.

1. INTRODUCTION

Business Process Management (BPM) is currently one of the most rapidly developing trends in management sciences. According to MH Jansen-Vullers and M. Netjes business process management supports processes using methods, techniques and software by designing, determining and analyzing operational processes, involving people, applications, documents and other sources of information [9]. Recent studies show that BPM has positive impact on organizational performance and supply chain collaboration [19] and observed trends in business practice suggest that the orientation on the business processes, their control and monitoring, is the current direction of modern organizations.

The literature on process methods with each year becomes more extensive and rich with new concepts and solutions. Starting from the historical M. Porter's value chain analysis, process reengineering [8], Activity Based Costing (ABC), Balanced Score Card (BSC) [11], Quality Management and Total Quality Management (TQM) concepts such as SixSigma. Nowadays, new methods of modeling, analysis and improvement of business processes become more and more popular.

Process mining techniques of collecting process data from existing events logs of IT systems [23] are especially useful in the context of creation of models of a process for future analysis. Many of these techniques, methods and tools are now a part of Business Process Management lifecycle (fig. 1). Process control and measurement stage of the process lifecycle is related to methods that provide guidance for the collection and consolidation of process related data [20]. These data can be further exploited for the process enhancement and optimization.

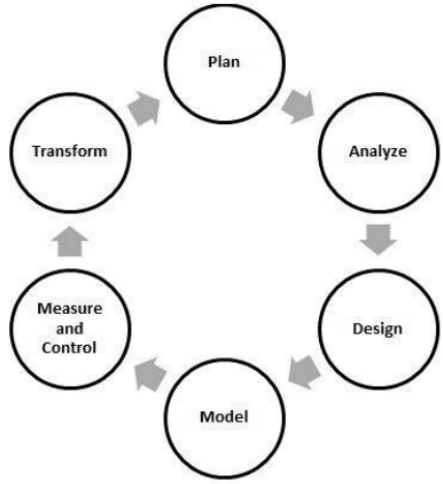

Fig. 1. Business Process Management lifecycle [21].

Predictive business process monitoring, a part of process mining, is a current new approach and

research area in BPM. Its aim is to combine machine learning and data mining algorithms with process mining techniques in order to predict the outcome of a process instance (such as the time remaining to completion of a case or whether the process instance will be violated or disrupted) or the next event in the process [3] based on its uncompleted traces and historical data. The results of such monitoring can be especially useful in the area of logistics and transport processes where process execution time, cost or required resources are essential for perfect process flow and customer satisfaction [5].

In this article we test a possibility of using tree-based classification algorithms for a prediction of a real-world process instance outcomes. Tree-based models were chosen because of their flexibility and insensitivity to extreme values (outliers) which are often present in real-world data [7]. The aim of the article is to evaluate the chosen algorithms as well as asses their value in the context of predictive business process monitoring.

The rest of the article is organized in the following way: in section 2 we describe the adopted methodology – the dataset, procedures of models' training and testing and adopted metrics of models' performance, section 3 presents the results, in section 4 we discuss the results and present related work, and section 5 contains final conclusions and ideas for the future.

2. METHODOLOGY

Here we simply describe our approach. For a particular task (activity) in the business process instance we try to predict if this task is going to be delayed. So the response variable is binary categorical variable with two values: *delayed* and *not delayed*. As predictor variables we take planned and actual execution times of the tasks that antecede the task which outcome is a subject of prediction. Some data are treated as "historical" and are used to train the classification model. A prediction is made based on the model and on the activities that has already occurred in the running process.

In the following subsections we describe in more details the data and the algorithms we have used and explain the metrics which were employed to assess the classification results.

2.1. DATASET

The dataset used in the study represents the actual business processes of the shipping company. Representation of this process includes a sequence of tasks (activities). The process starts with a start event and then a parallel run. Each of the executed transactions of the process from the start to the stop event can run in one (*i1*), two (*i1* and *i2*) or three (*i1*, *i2* and *i3*) incoming transport legs and one outgoing transport leg (*o*). Each of the transport legs contains four tasks (activities) which represent transport services. The transport services are described by three-letter acronyms according to the Cargo 2000 industry standard (Table 1).

Table 1. Types of tasks in the process and their description.

Task name	**Description of activities carried out in the task**
RCS	Check in freight at departure airline. Shipment is checked in and a receipt is produced at departure airport.
DEP	Confirm goods on board. Aircraft has departed with shipment on board.
RCF	Accept freight at arrival airline. Shipment is checked in according to the documents and stored at arrival warehouse.
DLV	Deliver freight. Receipt of shipment was signed at destination airport.

In addition, two of the four tasks in each leg (*DEP* and *RCF*) can be repeated up to 3 times (these repetitions are called hoops). The process is illustrated in Fig. 2 using the BPMN business process modeling standard.

A complex gateway is used in the diagram to separate parallel processes and to combine flows. Reconstruction of the process required the use of such gateway because there was no timestamp in the data. Repetitions in the process are modeled using the exclusion gateway for splitting and merging flows. Dataset includes 56,083 tasks related to 3,942 instances of process. Each task has assigned planned and actual execution time. More details about the data can be found in Metzger et al. [15].

Because of the different number of incoming transport legs and possible hoops, the number of tasks in the whole process (and, as a consequence, the number of predictor variables) is not constant. That is why we decided to treat each transport leg as a separate process. A visual exploratory analysis of the rate of delays of tasks in outgoing transport leg revealed that they do not depend on the number of incoming transport legs (fig. 3), so we believe that our approach is acceptable.

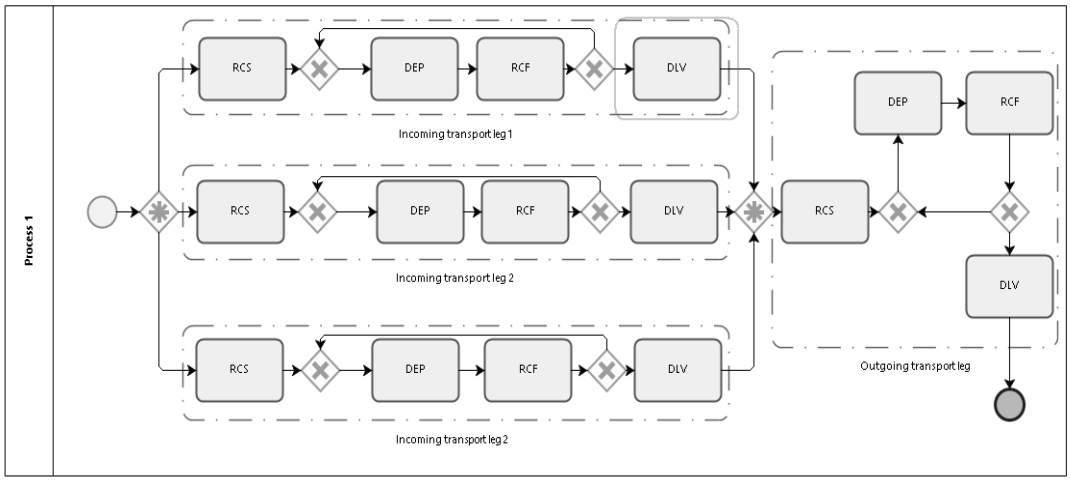

Fig. 2. Illustration of the process in BPMN.

The dataset was split into 5 smaller datasets, namely dataset1, dataset2, dataset3, dataset4 and dataset5. The number of each datasets indicates the number of tasks that happened before the task which outcome is the subject of the prediction[*]. For example, dataset2 means that the subject of the prediction is the third task in the transport leg (two tasks have already happened and their actual execution time is known when the predicti on is made).

many trees) usually give better results, but their interpretability is often limited. In view of this we chose four different tree-based classification models: single CART, single C5.0, random forest and extreme gradient boosting.

CART (classification and regression tree, very often described as decision tree) is probably the most famous tree-based algorithm. It tries to split the dataset into pure blocks (i.e. containing only one type of values of the response variable) based

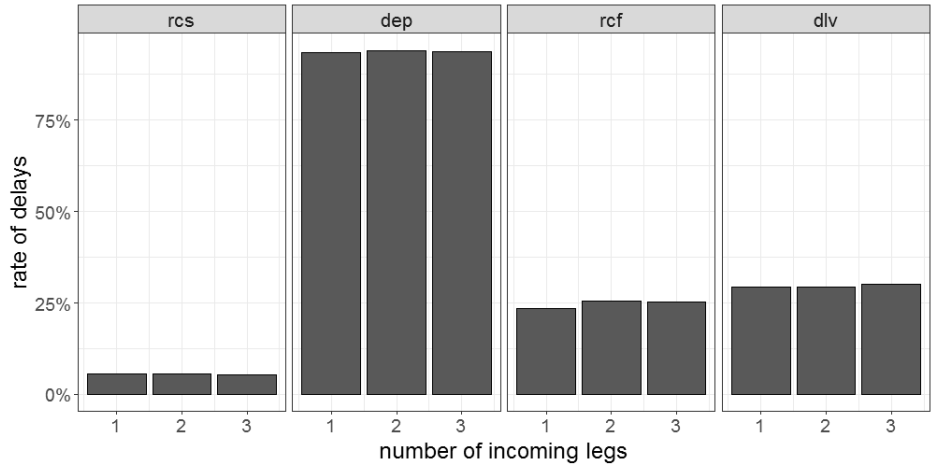

Fig. 3. Distributions of delays for the tasks in the outgoing transport leg with respect to the number of incoming transport legs.

2.2. METHODS

When using a tree-based classification algorithms one can encounter a trade-off between interpretability of the model and its actual performance. Single tree-based models are easy to understand and give clear guidelines, while ensemble models (i.e. consisting of a collection of

on Gini index or cross entropy. C5.0 is another classification tree-based algorithm which relies on information theory [12]. Random forest [1] consists of a collection of voting trees generated through random selection of input variables. In the study of Fernández-Delgado et al. [4] it received the best score among the 17 families of classifiers[*]. Apart from its high performance rate, random

[*] We removed from the dataset the transport legs with 3 hoops since they were extremely rare.

[*] Although recently there have appeared work discrediting these results [24].

forest is popular because of the small number of parameters that should be controlled. Extreme gradient boosting (xgboost) [2] belongs to a class of boosting methods, which iteratively modify weights (importance) of the voting trees in order to achieve the best accuracy. Gradient boosting algorithm can in some cases outperform random forest [17], but its many tuning parameters make it difficult to calibrate properly.

For each of the chosen algorithm and each of the five datasets we performed the following procedure.

1. A dataset was divided into training (67%) and testing (33%) datasets (the split was stratified, i.e. distribution of the response variable was preserved in both sets).
2. The model was trained on the training dataset using 10-fold cross-validation for parameter tuning. The training dataset can then be treated as "historical data" – the data representing transactions that have already ended and all of the values are known.
3. Performance of the model was checked on the testing dataset. This dataset represents the process runs that are not known.

This procedure is in accordance with the standard process of model testing in machine or statistical learning [7].

The tuning parameters of the models are presented in Table 2. The models were implemented in programming language R.

Table 2. Tuning parameters of the models.

Model	Search space	Type of search
CART	minsplit = {10, 20, 30}	grid
Random Forest	mtry = 1…number of predictors	grid
xgboost	nrounds = <50, 200> max_depth = <3, 10> min_child_weight = <1, 10> subsample = <0.5, 1> colsample_bytree = <0.5, 1>	random

2.3. METRICS

In binary classification problem (e.g. when trying to predict if an event such as delay occurs or not) one of the four situations is possible:

- true positive – an actual event is correctly predicted,
- true negative – a non-event is correctly predicted,
- false positive – an event is predicted but it does not occur,
- false negative – a non-event is predicted but an event actually occurs.

This is illustrated in Table 3 (called contingency table or confusion matrix). The table cells indicate the number of cases which were true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These values are used to calculate different metrics which characterize the performance of a classification algorithm.

Table 3. Contingency table.

		actual	
predicted		*event*	*non-event*
	event	TP	FP
	non-event	FN	TN

To assess the quality of the predictive algorithms we employ *accuracy* and Cohen's *Kappa* [12]. Accuracy is the simplest metric of classification algorithm – it measures the rate of correctly predicted cases. For example *accuracy* = 80% means that only 20% of all cases were incorrectly classified. However, using only accuracy to assess the model can be misleading since high accuracy can be obtained in the case of highly imbalanced frequencies of response variable classes. That is why we also use Kappa statistic which takes into account these classes distributions and is calculated using observed and expected accuracy. As Kuhn and Johnson [12] point out, Kappa values greater than 0.3 indicate quite reasonable agreement between predicted and observed values.

These two metrics however do not make any distinction between error types that can be made. Following Metzger et al. [15] we also consider three more metrics.

- *Precision* – it measures how many predicted delays were actual delays; higher value of precision means smaller rate of false alarms (i.e. incorrectly predicted delays).
- *Recall* – it indicates the rate of actual delays classified correctly; the higher the value of recall, the more actual delays are recognized.
- *F1* – it combines precision and recall.

All of the metrics and their formulas are presented in Table 4.

Table 4. Performance metrics and their formulas.

Metric name	Formula
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$
Kappa	$\frac{Accuracy-E}{1-E}$ where: $E=\frac{(TP+FP)(TP+FN)+(FP+TN)(FN+TN)}{(TP+FP+FN+TN)^2}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1	$\frac{2\cdot precision\cdot recall}{precision+recall}$

3. RESULTS

In fig. 4 *accuracy* and *Kappa* statistic are presented. *Precision*, *recall* and *f1* are in fig. 5. Table 5 contains the values of all performance metrics for each model and each dataset.

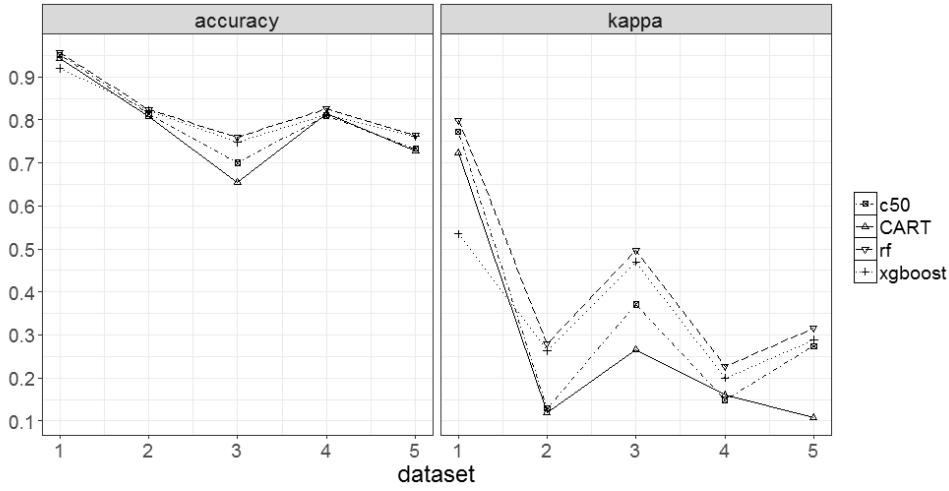

Fig. 4. Accuracy and Kappa indicators of models.

The accuracy of almost all models (with the exception of CART in dataset3) exceeded 70%. This means that in most cases no more than 30% of the tasks were predicted incorrectly. For each dataset the highest accuracy was achieved by random forest, but xgboost was almost equally good for datasets 3 and 5 (see Table 5 for the precise values). However, as it was mentioned earlier, accuracy is not always the best indicator of the predictive power of the model. In the right panel of fig. 4 Kappa statistics is presented. As it can be seen, the order of the algorithms is preserved, with random forest having the highest values and xgboost being in most cases on the second place. But the values indicate that the models' performance is not as good as it would suggest the accuracy measure. In particular, random forest exceeded the 0.3 threshold in three datasets, xgboost and C5.0 in two, and CART only in one dataset. Datasets 2 and 4, despite high accuracy, were the most problematic for the tested algorithms. This is also seen in figure 5, especially in case of recall and f1 metrics. The values of recall for the random forest indicate, that in *dataset2* only about 25% of actual delays were correctly recognized, and less than 20% in *dataset4*. The other models achieved even lower scores. The results are much better in case of precision. All values exceed 50% which means that whenever a model predicts delay, in most cases it is a correct prediction. It is worth mentioning that random forest is not always the most "precise" algorithm – C5.0 is better in *dataset2* and xgboost in *dataset5*.

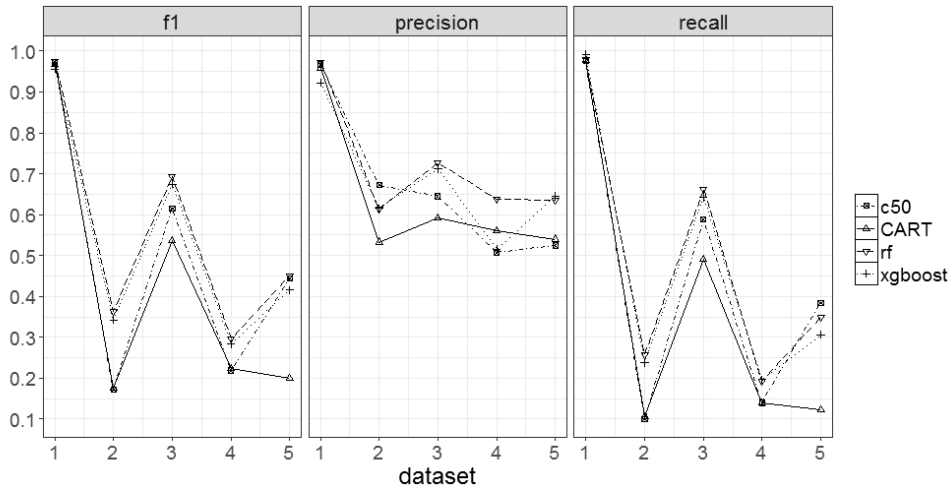

Fig. 5. F1, precision and recall of models.

Table 5. Models' performance metrics.

dataset	model	accuracy	kappa	precision	recall	f1
1	CART	0.9431	0.7243	0.9577	0.9781	0.9678
2	CART	0.8070	0.1195	0.5329	0.1057	0.1765
3	CART	0.6553	0.2662	0.5929	0.4903	0.5367
4	CART	0.8154	0.1598	0.5606	0.1396	0.2236
5	CART	0.7284	0.1082	0.5402	0.1221	0.1992
1	C5.0	0.9512	0.7719	0.9673	0.9772	0.9722
2	C5.0	0.8144	0.1291	0.6726	0.0992	0.1729
3	C5.0	0.7005	0.3711	0.6449	0.5887	0.6155
4	C5.0	0.8103	0.1490	0.5068	0.1396	0.2189
5	C5.0	0.7335	0.2740	0.5248	0.3844	0.4438
1	random forest	0.9569	0.7996	0.9716	0.9793	0.9754
2	random forest	0.8231	0.2792	0.6137	0.2572	0.3625
3	random forest	0.7605	0.4970	0.7261	0.6614	0.6923
4	random forest	0.8254	0.2274	0.6375	0.1925	0.2957
5	random forest	0.7644	0.3156	0.6351	0.3481	0.4497
1	xgboost	0.9201	0.5347	0.9228	0.9915	0.9559
2	xgboost	0.8218	0.2624	0.6149	0.2376	0.3427
3	xgboost	0.7483	0.4704	0.7119	0.6414	0.6748
4	xgboost	0.8118	0.2001	0.5149	0.1962	0.2842
5	xgboost	0.7615	0.2887	0.6448	0.3065	0.4155

For better evaluation of the obtained results we carried out additional analysis and compared the values of accuracy of the two best predictors (random forest and xgboost) with a "dummy predictor" which simply returns the most probable outcome. This analysis was conducted to check if the information gained from the models exceeded significantly simple random guessing. Fig. 6 presents the values and in Table 6 there are results of statistical comparison of population proportions between models and dummy classifier. As a sampling size the number of cases in the testing datasets was assumed.

In each dataset the results of models are better than the accuracy of the dummy classifier. Statistical significance (p-value < .05) was observed in four of the five datasets. This means that for these datasets algorithmic predictions are indeed better than simple random guessing based on the historical distribution of the delays. In *dataset4* the differences between dummy classifier and models' predictions were too small to treat them as significant from the statistical inference point of view.

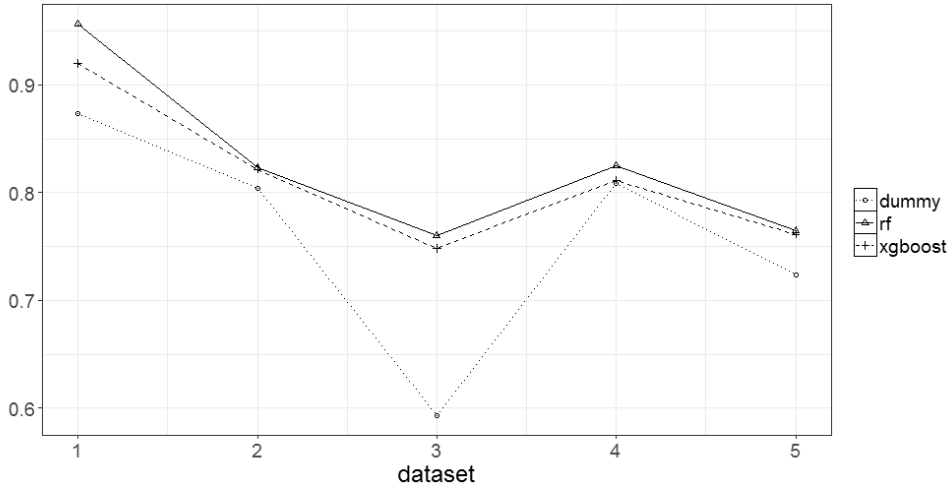

Fig. 6. Accuracies of random forest, xgboost and dummy predictor.

Table 6. Statistical comparisons of random forest and xgboost accuracies with the dummy classifier.

dataset	dataset size (testing)	dummy	random forest		xgboost	
		accuracy	accuracy	p-value	accuracy	p-value
1	3917	0.8737	0.9569	< .001	0.9201	< .001
2	3917	0.8044	0.8231	0.0032	0.8218	0.0060
3	3917	0.5927	0.7605	< .001	0.7483	< .001
4	1392	0.8091	0.8254	0.1210	0.8118	0.7987
5	1392	0.7234	0.7644	< .001	0.7615	0.0015

4. DISCUSSION AND RELATED WORK

The results presented in the previous section show that ensemble methods (random forest, xgboost) perform better than single trees, which was expected. However the differences between models accuracies are not very big. Larger diversification can be observed in case of the other metrics, especially Kappa, f1 and recall. Overall, random forest turned out to be the best algorithm (although it did not always achieve the highest metrics values), with extreme gradient boosting being the second best. It is quite possible that with more parameter tuning the results of xgboost could be better.

What is interesting is the fact that the best scores were achieved for *dataset1* - the one with the smallest number of predictors. This means that the first two tasks in the process (*RCS* and *DEP*) are strictly connected with each other. Another thing worth mentioning is that models' performance does not increase with the process execution time. Actually the worst results were obtained for *dataset4*, so it can be concluded that the information gained from activities that has already occurred in the running process is not always very helpful in order to predict the outcome of the next activity. In their analysis of the same dataset Metzger et al. [15] actually attained quite different result. But they divided the data in a different way so it is impossible to compare the effects.

In the literature different approaches to predict the binary outcome of the process instance can be found. Single decision trees were used by Maggi et al. [14] in order to predict violations of business goals defined in the form of linear temporal logic rules. Leontjeva et al. [13] used random forest and compared its performance with support vector machine and generalized boosted regression models. Random forest combined with logistic regression were also used by Teinemaa et al. [22]. Clustering methods were used by Folino et al. [6] in order to predict violations in service level agreement terms and by Kang et. al. [10] to detect abnormal termination. An interesting approach was presented in Metzger et al. [15] where three technics: neural networks, constraint satisfaction and Quality-of-Service aggregation were combined.

5. CONCLUSIONS

In this article we presented the application of tree-based classification algorithms (CART, C5.0, random forest and extreme gradient boosting) in order to predict the outcome of a process instance. We trained and checked these models on real-

world event-log data. The results suggest that from the four tested algorithms random forest gains the best scores according to the evaluation metrics we employed (accuracy, Cohen's Kappa, precision, recall, and f1). The obtained results demonstrated however that the knowledge about the activities that has already occurred in the current process may not be enough to predict the outcome of the succeeding task. A possible solution is to include as predictors additional data sources, since many of the disruptions in logistic processes arise from external factors [16]. Another possibility is in-depth process data exploration [18], as it can give additional insights about the data and relationships between variables and in this way improve the results of prediction.

REFERENCES

[1] Breiman L., 2001. Random forests. *Machine learning*, 45, 5-32.

[2] Chen T., Guestrin C., 2016. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM International Conference On Knowledge Discovery And Data Mining*, 785-794.

[3] Evermann, J., Rehse, J. R., Fettke, P., 2016, A deep learning approach for predicting process behaviour at runtime. *International Conference on Business Process Management*, Springer, Cham, 327-338.

[4] Fernández-Delgado M., Cernadas E., Barro S., Amorim D., 2014. Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*, 15, 3133-3181.

[5] Folinas D., Bochtis D., Sorensen C., 2010. In-field logistics processes management based on business activities monitoring systems paradigm. *International Journal of Logistics Systems and Management*, 8, 1–18.

[6] Folino, F., Guarascio, M., Pontieri, L., 2012. Discovering context-aware models for predicting business process performances. In: Meersman, R., Panetto, H., Dillon, T.,Rinderle-Ma, S., Dadam, P., Zhou, X., Pearson, S., Ferscha, A., Bergamaschi, S., Cruz, I.F. (eds.) *OTM 2012, Part I. LNCS*, vol. 7565, Springer, Heidelberg, 287–304.

[7] Friedman J., Hastie T., Tibshirani R., 2009. *The elements of statistical learning*. New York: Springer series in statistics.

[8] Hammer M., Champy J., 1996. Reengineering w przedsiębiorstwie. Neumann Management Institute, Warszawa.

[9] Jansen-Vullers M.H., Netjes M. Business Process Simulation - A Tool Survey. *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, 38, Aarhus, Denmark, 1-20.

[10] Kang, B., Kim, D., Kang, S., 2012. Real-time business process monitoring method for prediction of abnormal termination using knni-based LOF prediction. *Expert Syst. Appl.* 39, 6061-6068.

[11] Kaplan R. S., David P. Norton, 1992. The Balanced Scorecard – Measures that Drive Performance. *Harvard Business Review*, 71-79.

[12] Kuhn M., Johnson K., 2013. *Applied predictive modelling*. New York: Springer.

[13] Leontjeva, A., Conforti, R., Francescomarino, C.D., Dumas, M., Maggi, F.M., 2015. Complex symbolic sequence encodings for predictive monitoring of business processes. *Business Process Management - 13th International Conference, BPM 2015*, Innsbruck, Austria, 297-313.

[14] Maggi, F. M., Di Francescomarino, C., Dumas, M., Ghidini, C., 2014. Predictive monitoring of business processes, *International Conference on Advanced Information Systems Engineering*, Springer, Cham, 457-472.

[15] Metzger A., Leitner P., Ivanović D., Schmieders E., Franklin R., Carro M., Dustdar S., Pohl, K., 2015. Comparing and combining predictive business process monitoring techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45, 276-290.

[16] Nowakowski, T., Werbinska-Wojciechowska, S., Chlebus, M.,2015. Supply Chain Vulnerability Assessment Methods–Possibilities and Limitations. *Safety and Reliability of Complex Engineered Systems*, Taylor & Francis Group, London,1667-1678.

[17] Ogutu J. O., Piepho H-P., Schulz-Streeck T., 2011. A comparison of random forests, boosting and support vector machines for genomic selection. *BMC proceedings*, 5, BioMed Central, 2011, available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3103196/

[18] Owczarek T., 2017, An Example of Exploratory Analysis for Predictive Business Process Monitoring. *Proceedings of the 30th International Business Information Management Association Conference*, Madrid, Spain, 4224-4228.

[19] Pradabwong, J., Braziotis, C., Tannock, J., & Pawar, K. S., 2017. Business process management and supply chain collaboration: effects on performance and competitiveness. *Supply Chain Management: An International Journal*, 22(2).

[20] Rosemann M., vom Brocke J., 2015. The Six Core Elements of Business Process Management. *Handbook on business process management, vol 2.*, Springer Heidelberg, 105-122.

[21] Ruzevicius J., Miškele M., Darius K., 2012. Peculiarities of The Business Process Management Lifecycle at Different Maturity Levels: The Banking Sector's Case. *Issues of Business and Law,* 4, 2012.

[22] Teinemaa, I., Dumas, M., Maggi, F. M., Di Francescomarino, C., 2016. Predictive business process monitoring with structured and unstructured data. *International Conference on Business Process Management*, Springer International Publishing, 401-417.
[23] van der Aalst W. M. P., M. H. Schonenberg, Song M., 2011., Time prediction based on process mining. Information Systems, 36, 450–475.
[24] Wainberg M., Alipanahi B., Frey B. J., 2016. Are random forests truly the best classifiers? *The Journal of Machine Learning Research*, 17, 3837-3841.

Date submitted: 2018-07-17
Date accepted for publishing: 2018-10-31

Tomasz Owczarek
Silesian University of Technology, Poland
tomasz.owczarek@polsl.pl