UDC 330.341.1 https://doi.org/10.33271/ebdut/90.185

PREDICTIVE MONITORING OF BUSINESS PROCESSES WITHIN THE FRAMEWORK OF DIGITAL ECONOMY DEVELOPMENT

T. I. Mshvidobadze, Professor, Gori State University (Georgia),
tinikomshvidobadze@gmail.com, orcid.org/ 0000-0003-3721-9252

Methods. In this paper, we present an approach to analyze such event logs in order to predictively monitor business constraints during business process execution. At any point during an execution of a process, the user can define business constraints in the form of linear temporal logic rules. The framework defines the input data values that are more or less likely to meet each business constraint. The approach has been implemented in the ProM process mining toolset.

Results. The results show that this model can provide competitive classification performance, create highly interpretable models, and effectively reduce data preparation efforts. The calculation of the information gain coefficient criteria is performed and shown in the algorithm using the appropriate equations and a recursive function. In this paper, we describe J48SS, a new decision tree inducer based on the C4.5 algorithm. The algorithm is experimentally validated on a real business speech analytics setting. J48SS is shown to effectively reduce the data preparation effort, and the use of binary splits shows that the tree grows fully. Future work could be devoted to investigating ensembles of J48 trees and applying the proposed algorithm to a corresponding database, where all types of supported attributes naturally arise. Also, a future research direction could be to extend the model to deal with temporal logic formulas. Such a formulation would allow the decision tree to take into account relationships between the values of different attributes, instead of looking at each of them individually.

Novelty. This paper presents J48SS, a new decision tree learner that can blend static, sequential, and time-series data for classification purposes. The new algorithm is based on the popular C4.5 decision tree learner and relies on the concepts of frequent pattern extraction and time-series formula generation. A framework for predictive business process monitoring is shown.

Practical value. In this article, we propose implementation of the MapReduce C4.5 algorithm. Empirical results indicate that the implementation of the algorithm demonstrates both time efficiency and scalability.

Keywords: digital economy, business process modeling, process monitoring, C4.5 algorithm.

Statement of problem. The phenomenon of the development of the digital economy has been examined in a number of works by scientists. It is need to highlight Brynjolfsson (Brynjolfsson, 2000), who made an attempt to study the digital economy in depth, identified the digital economy as a driving force for economic development, which gives the potential to make significant economic changes, influencing on various spheres of activity, level of labor and the very way of human life.

A broad study of the digital economy and digitalization processes was found out by Dahlman et al, (Dahlman et al, 2019), in which it is determined that the emergence of new technologies reorganizes existing economic processes, which leads to a change in the economic processes themselves, the reorganization of economic systems;

This circumstance has a more positive effect on developing countries. Manyika et al, (17, pp. 41–47) approached the issue of digitalization from a practical point of view, determining that data flows are a serious component of the global economy, along with trade flows.

These works are only a small part of the whole set of works devoted to the concept, development and problems of the digital economy. Today it becomes obvious that the world economic system is shifting towards a new – digital economy, and, therefore, a more in-depth and detailed study of this process is needed.

The goal of the study is to determine the importance of the digital economy today and monitor the implementation of business processes.

Based on the goal, we define the main tasks of the study:

• identification of key competencies for the development of the digital economy and identification of problems and shortcomings;

• analysis of the transformation of business processes in the digital economy and the proposal of business process models for various fields of activity.

The execution of business processes is generally subject to internal policies, norms, best practices, regulations, and laws.

Compliance monitoring is an everyday imperative in many organizations. Accordingly, a range of research proposals have addressed the problem of monitoring business processes with respect to business constraints. Given a process model and a set of constraints – expressed, e.g., in temporal logic – these techniques provide a basis to monitor ongoing executions of a process in order to assess whether they comply with the constraints in question.

In this setting, this paper presents a novel monitoring framework, namely Predictive Business Process Monitoring, based on the continuous generation of predictions and recommendations on what activities to perform and what input data values to provide, so that the likelihood of violation of business constraints is minimized. At any point during the execution of a business process, the user can specify a business constraint using Linear Temporal Logic (LTL). Based on an analysis of execution traces, the framework continuously provides the user with estimations of the likelihood of achieving each business constraint for a given ongoing process execution. The proposed framework takes into account the fact that predictions often depend both on: (i) the sequence of activities executed in a given case, and (ii) the values of data attributes after each activity execution in a case. The proposed framework can be applied both for prediction and recommendation.

This paper presents J48SS, a novel decision tree learner based on WEKA’s J48 (a python implementation of C4.5). The algorithm is capable of naturally exploiting categorical, numerical, sequential, and time series data during the same execution cycle. The resulting decision tree models are intuitively interpretable, meaning that a domain expert may easily read and validate them.

Since multiple algorithms have to be combined to produce a final classification, the final model may lack in interpretability. This is a fundamental problem in all domains in which understanding and validating the tranined models is as important as the accuracy of the classification itself, e.g., production business systems and life critical medical applications.

For prediction, the decision tree is used to evaluate the probability for the business constraint to be satisfied for a given combination of attribute values. For recommendation, the decision tree is used to select combinations of attribute values that maximize the probability of the business constraint being satisfied. The predictive monitoring framework has been implemented in the ProM toolset for process mining. ProM is unique due to the purposeful use of its features.

Analysis of recent papers. The most complete approaches and evolution of the term «digital economy» were considered by Buht and Hicks in their work «Definition, Concept and Measurement of the Digital Economy» (5, pp. 143–172).

Therefore, most researchers believe that in Don Tapscott's work «Digital Economy», the author himself meant a certain transition from the traditional economic model to a new one, taking into account the development of new technologies, innovations, and digitalization, which gave the name to the new economic model.

A number of authors have indicated in the list of shortcomings or inaccuracies in the development and development of digitalization programs (11). In their opinion, one of the main shortcomings is the lack of an adequate system for assessing the growth and pace of development of the digital economy.

The digitalization of business provides a vast amount of data that can be used with modern methods of business process management. Recently, researchers have been trying to incorporate additional process-related information, also known as process context, into their predictive models.(26, pp. 43-47).

Despite the sufficient diversity of approaches to the understanding and development of the digital economy, researchers around the world demonstrate a certain unanimity in the area of key competence of the digital economy, which, in their opinion, is data analysis. Geissbauer et al, provides a diagram for the development of Industry 4.0 and related technologies which indicates that data analysis is the basis for the development of all digitalization processes: value chain integration, business modeling and user access, goods and services in the digital economy (13, p.57). Most studies do not call data analysis a key competency, but imply this, based on the logic of their research. For example, Anand (1, pp.45-60) formulates three main trends in the development of modern digital enterprises based on data analytics. Aspects of data analysis in the modern digital economy are reflected in the OECD country report (OECD).

At the same time, all the listed authors note both global progress in the development of digital technologies and the importance of the transition to a new type of economic relations based on a digital basis.

There are several papers in the literature that provide approaches for generating predictions and recommendations during process execution and are oriented towards a time perspective.

Aalst et al. presented a set of approaches based on annotated transition systems that contain time information extracted from event logs.(2, pp. 38–52).

Conforti et al. presented a process support technique to help participants make risk-informed decisions to reduce process risks. Risks are predicted by traversing decision trees generated from past process execution logs. (7, pp. 116–132).

Pika et al. proposed an approach for time-related prediction, by identifying observable indicators in event logs that highlight the possibility of missing deadlines.(21, pp. 211–216).

Suriadi et al. proposed an approach for Root Cause Analysis based on classification algorithms. After enriching the log, such as workload, delay, and engagement events from resources, they use decision trees to determine the causes of overtime errors. (24, pp. 174–186).

An approach to predicting abnormal termination of business processes is presented by Kang, et al. Here, a fault detection algorithm is used to estimate the probability of fault occurrence. Alarms are provided for early notification to avoid potential risks. (15, p.120).

Castellanos et al. presented a business operations management platform equipped with time series forecasting capabilities. This platform allows for forecasting metric values on current process instances, as well as for predicting aggregated metric values for future instances.

Predictive Process Monitoring is a branch of process mining that aims at predicting the future of an ongoing (uncompleted) process execution. Typical examples of predictions of the future of an execution relate to the outcome of a process execution, to its completion time, or to the sequence of its future activities.

Prescriptive Process Monitoring systems aim at recommending, during the execution of a business process, interventions that, if followed, prevent poor performance of the process.

One of the main trends in Industry 4.0 refers to predictive monitoring and recommendations based on the data coming from information systems, machines, and IoT sensors. Predictive monitoring allows users to perform cost-effective interventions and determine them ahead of time before a (costly) system failure/process negative outcome occurs. On average, predictive monitoring (and maintenance) increases productivity by 25%, reduces failures by 70% and lowers costs by 25%. (9, pp. 28–35). However, with the increasing development of cloud computing as well as the big data challenge (14, pp. 47–50).

Traditional decision tree algorithms exhibit multiple limitations. First and foremost, building a decision tree can be very time consuming when the volume of dataset is extremely big, and new computing paradigm should be applied for clusters. Second, although parallel computing in clusters can be leveraged in decision tree based classification algorithms (4, pp. 188–189), the strategy of data distribution should be optimized so that required data for building one node is localized and meanwhile the communication cost of minimized. To this end, in this paper we demonstrate a distributed implementation of the C4.5 algorithm.

Various aspects of decision trees have been addressed by many researchers. The best part of the decision trees is that the presence of noise doesn't hamper the performance of the classifier. Likewise, the accuracy of a decision tree isn't affected by the presence of redundant attributes. (20, pp. 323–332). This paper presents J48SS, a novel decision tree learner based on WEKA's J48 (a python implementation of C4.5). The algorithm is capable of naturally exploiting categorical, numerical, sequential, and time series data during the same execution cycle. The resulting decision tree models are intuitively interpretable, meaning that a domain expert may easily read and validate them.

Materials and methods. This article uses descriptive, analytical, and explanatory methods, based on which important issues of the study are highlighted. To formulate the concept, we examined the views of various researchers, based on which we illustrated the reasoning and conclusions.

In this paper, we describe J48SS, a new decision tree inductor based on the MapReduce C4.5 algorithm. The new approach can efficiently decode sequential and time series data within the same execution cycle to proactively observe business constraints during the execution of a business process. The presented J48SS, a decision tree framework, is based on J48 (WEKA's implementation of C4.5).

To extract meaningful information from sequential data, J48SS relies on the VGEN frequent closed pattern extraction algorithm (12, pp. 476–488). All changes that were made to integrate the two algorithms seamlessly are thoroughly described. The relevant data are obtained as a result of experiments.

Running Example. During the execution of a business process, process participants cooperate to satisfy certain business constraints. At any stage of the process enactment, decisions are taken aimed at achieving the satisfaction of these constraints. Therefore, it becomes crucial for process participants to be provided with predictions on whether the business constraints will be achieved or not and, even more, to receive recommendations about the choices that maximize the probability of satisfying the business constraints.

The approach aims at supporting process participants in their decisions by providing them with predictions about the satisfaction of their constraints and, in case they can influence the process with their decisions, by recommending them the best choices to be made to satisfy their business constraint.

We discuss the language used to define business constraints (LTL) and then we give an overview of decision tree learning.

In the proposed approach, business constraints can be formulated according to LTL rules, as LTL (and its variations) are classically used in the literature to express business constraints in procedural knowledge (19, pp. 287–300), LTL is a modal logic with modalities dedicated to describing aspects of time.

Decision tree learning uses a decision tree as a model to predict the value of a target variable based on input variables (functions). Decision trees are built from a training data set. Each leaf of the decision tree is labeled with a class, or value of the target variable, given the values of the input variables, which are represented by a path from the root to the leaf.

Each leaf of the decision tree is associated with class support and a probability distribution. Class support is the number of examples in the training set that follow the path from the root to the leaf and are correctly classified; class probability (prob) is the percentage of examples correctly classified relative to all examples that follow that path, as shown in the formula (1):

$$Prob = \frac{\mathrm{N}(corr\ class\ leaf\ examples)}{\mathrm{N}(\text{corr class leaf examples + incorr class leaf examples})} \quad (1)$$

Fig. 1. sketches the proposed Predictive Business Process Monitoring Framework. It relies on two main modules: a Trace Processor module to filter and classify (past) execution traces and a Predictor module, which uses the Trace Processor output as training data to provide predictions and recommendations.

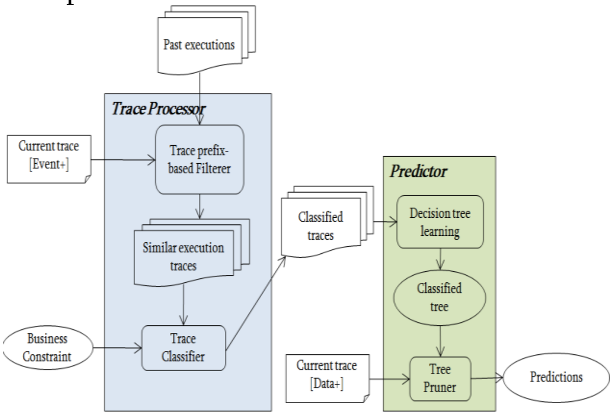

Fig. 1. Sketches the proposed Predictive Business Process Monitoring Framework.

The Trace prefix-based Filterer submodule of the Trace Processor module extracts from the set of historical traces only those traces having a prefix control flow like the one of the current execution traces. Filtering is needed since data values are usually strongly dependent on the control flow path followed by the specific execution. In addition, traces with similar prefixes are more likely to have, eventually in the future, a similar behavior. The similarity between two traces is evaluated based on their editing distance.

This abstraction is used to guarantee that there are enough examples to use for learning a decision tree. The traces (data snapshots) of the training set are classified by the Trace Classifier submodule according to whether each of them satisfies the desired business constraint. The constraint is expressed in terms of a set of LTL formulas.

Once the corresponding traces, and therefore the corresponding data snapshots, have been classified, they are passed to the decision tree training module, which is responsible for assigning the learned decision tree to the associated class support and probability.

All the data values assigned in the past, are supposed to be known by the predictor system at the current execution point of the trace. The tree can hence be pruned by removing all the branches corresponding to known values. The pruning algorithm returns either a unique path or a subtree of the original tree, according to whether the system is used as predictor (the values of all the tree attributes are known) or as a recommender respectively.

In the latter case, leaves are ranked according to the associated class probability. The conditions on the values of the unknown attributes corresponding to the leaves with the highest rankings are returned to the user as recommendations.

Therefore, the standard C4.5 algorithm is used to induce classification rules in the form of a decision tree. The C4.5 algorithm (J48) is mostly used among many fields for classifying data for example interpreting the clinical data for the diagnosis of coronary heart disease, classifying E-governance data, and many more. The C4.5 algorithm, compared to Random Forest and other powerful machine learning algorithms that are widely used for classification and regression tasks, has an overfitting algorithmic technique, with performance, flexibility, and parameter tuning. It is one of the most widely used algorithms for learning decision trees (22, p.37), which relies on normalized information retrieval to select a function for each node in the tree, which is used to split a set of examples. The function is selected by retrieving the most normalized information to make a decision. It is an extension of Ross Quinlan's earlier ID3 algorithm also known in Weka as J48.

C4.5 algorithm is one of the most widely used machine learning algorithms to examine the data categorically and continuously. The C4.5 algorithm (J48) is mostly used among many fields for classifying data for example interpreting the clinical data for the diagnosis of coronary heart disease, classifying E-governance data, and many more.

As an extension of ID3 (23, pp. 78–82), the default criteria for choosing splitting attributes in C4.5 is *information gain ratio*. Instead of using information gain as that in ID3, the information gain ratio avoids the bias of selecting attributes with many values. The Python programming language emphasizes code simplicity thereby enabling programmers to develop applications quickly. (6, p.87).

Algorithm 1 C4.5(T)

```
Input: training dataset T; attributes S.
Output: decision tree Tree.
1: if T is NULL then
2:     return failure
3: end if
4: if S is NULL then
5:     return Tree as a single node with most frequent class label in T
6: end if
7: if |S| = 1 then
8:     return Tree as a single node S
9: end if
10: set Tree = {}
11: for a ∈ S do
12:     set Info(a,T) = 0, and SplitInfo(a,T) = 0
13:     compute Entropy(a)
14:     for v ∈ values(a,T) do
15:         set T_{a,v} as the subset of T with attribute a = v
16:         Info(a,T)+ = |T_{a,v}|/|T_a| Entropy(a_v)
17:         SplitInfo(a,T)+ = -|T_{a,v}|/|T_a| log |T_{a,v}|/|T_a|
18:     end for
19:     Gain(a,T) = Entropy(a) - Info(a,T)
20:     GainRatio(a,T) = Gain(a,T)/SplitInfo(a,T)
21: end for
22: set a_best = argmax_a {GainRatio(a,T)}
23: attach a_best into Tree
24: for v ∈ values(a_best,T) do
25:     call C4.5(T_{a,v})
26: end for
27: return Tree
```

Fig. 2. C4.5 Algorithm Description

If we denote the number of classes C, and p (S, j) is the proportion of cases in S that are assigned to class j. Thus, the entropy of attribute S is calculated as follows:

$$\text{Entropy (S)} = -\sum_{j=1}^{C} \mathrm{P(S,J)} \times \log \mathrm{P(S,J)} \qquad (2)$$

While, the information gain coefficient of attribute S is defined as:

$$\text{GainRatio (S,T)} = \frac{Gain\ (S,T)}{\text{SptInfo(S,T)}} \qquad (3)$$

Where $\mathrm{SptInfo(S,T)}$ is calculated as:

$$\mathrm{SptInfo(S,T)} = -\sum_{v \in Value\ T_s} \frac{[T_{s,v}]}{T_s} \times log \frac{[T_{s,v}]}{T_s} \qquad (4)$$

Thus, the whole process of C4.5 algorithm is described in Algorithm 1. The information gain ratio criteria computation is performed in lines 11~21 using above equations, and a recursive function call is done in Line 25.

Decision Tree Learning Implementation. There are many environments for implementing algorithms, for example, the ProM process mining toolkit. ProM provides a generic operating support (OS) environment (2, pp. 38–52), that allows the tool to interact with external workflow management systems. The event stream from the workflow management system is received by the OS service. The OS service is connected to OS providers that perform various types of analysis that can be performed online in the stream.

Fig. 3. shows the overall architecture. The OS service receives the event stream (including the current execution trace) from the workflow management system and sends it to a predictive business process monitoring framework, which returns predictions and recommendations. The OS service sends these results to the workflow management system (25, pp. 169–188).

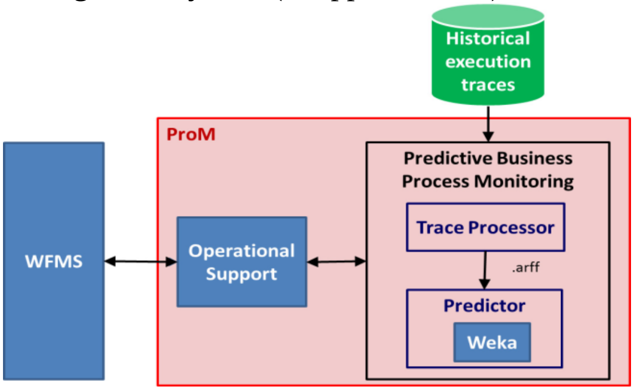

Fig. 3. Predictive Business Process Monitoring Framework: implemented architecture

Predictor is effectively implemented using the WeKa J48 implementation of the C4.5 algorithm.

The J48 implementation of the C4.5 algorithm has many additional features including accounting for missing values, decision trees pruning, continuous attribute value ranges, derivation of rules, etc. In the WEKA data mining tool, J48 is an open-source Java implementation of the C4.5 algorithm. J48 allows classification via either decision trees or rules generated from them, which takes an .arff file as input and builds a decision tree. The .arff file contains a list of typed variables and the corresponding values for each trace prefix (i.e., each data snapshot). This file is created by the Trace Processor and passed to the Predictor. The resulting decision tree is then analyzed to generate predictions and recommendations. (28, pp. 141–149).

Weka window consists of various classifiers like bays, functions, lazy, meta and tree etc. available in weka. First click on trees, then choose J48 (c4.5 is termed as J48 in weka software) which results in following figure 4. Decision tree growth can also be done in data mining using the c4.5 algorithm of data classification techniques.

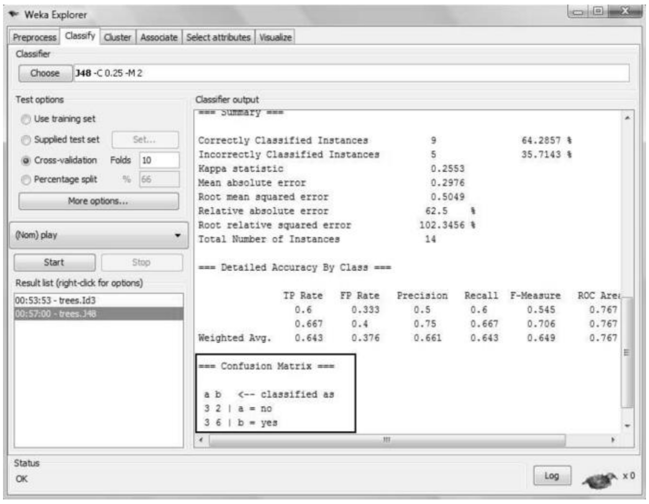

Fig. 4. Weka run information for C4.5

The name of the classifier is listed in the text box right beside the choose button. The text after j48 represents parameter setting. These are default parameters which state that the confidence factor for pruning is 0.25, to use binary splits and restrict the minimum no. of instances in a leaf to 2 which means grow the tree fully. The experiments were conducted using the BPI Challenge event log. This log refers to a healthcare process and contains process executions related to the treatment of patients.

The entire event log contains 1,143 cases and 150,291 events, which are distributed across 623 event classes (activities). Each case refers to a different patient treatment. The event log contains domain-specific attributes that are both case attributes and event attributes, in addition to the standard XES attributes. For example, age, diagnosis, and treatment code are case attributes, and activity code, number of performances, specialty code, and group are event attributes.

In the experiments, first, traces were placed in the log based on the time at which the first event of each trace occurred. Then, these traces were used as historical data to derive predictions. To simulate the trace execution (remaining 20%) they were sent as an event stream to the OS service in ProM (test suite).

Within the framework of the experiments, 5 business constraints were defined, which correspond to a subset of LTL rules. These rules apply to all LTL constructs, while real business constraints are investigated.

In the first experiment, a similarity threshold of 0.8 and a minimum number of traces of 30 were used. ROC space analysis was used to evaluate the effectiveness of the approach.

ROC space analysis highlights that the OS provider was able to distinguish well between positive and negative results.

In the second experiment, we use a low similarity threshold (0.5) and again, a minimum number of traces of 30. This experiment shows that considering only predictions with high class probability does not always improve the quality of the results, although the percentage of missing predictions is not high (about 20%).

In summary, the evaluation shows that the proposed approach is feasible and provides accurate predictions (and hence recommendations). Results seem overall not to be affected by the position of the evaluation point, thus demonstrating that the approach works well even when few variables are known.

Conclusions. This paper presents a framework for predictive business process monitoring based on the estimation of the probability of LTL rule execution at different stages of case execution. The framework considers both the sequence of activities and the data related to the execution of each activity. Validation of the framework using real logs shows that the recommendations based on the framework have a promising level of accuracy when sufficient support is available.

There are several works in the literature that provide a set of approaches based on annotated transient systems that contain timing information extracted from event logs.

The approach proposed in the paper aims to maximize the probability of satisfying business constraints, which are expressed in the form of LTL rules.

Increased accuracy can be achieved by extending the technique in two directions. First, the proposed technique matches the current case trace with the prefixes of the completed trace based on the edit distance. Similarly, discriminant sequence mining techniques can be used to extract prefix patterns that are associated with the fulfillment of a given business constraint. (16, pp. 21–32). Secondly, we have considered the use of decision trees to build a classifier. In the case of a larger number of attributes, which can be encountered in richer logs, decision trees have limitations and are likely to show low accuracy due to their inherent weaknesses when dealing with large feature sets. In this context, other classification techniques, such as random forests or sparse logistic regression, are recommended.

In this paper, we described J48, a novel decision tree inducer based on the C4.5 algorithm. The novel approach is capable of effectively mixing, during the same execution cycle, static, as well as sequential and time series data. The resulting models are highly readable, which is a fundamental requirement in those domains in which understanding the classification process is as important as the classification itself, for instance for validation purposes.

The algorithm was tested by experimenting on a real business speech analytics setting. J48SS has been shown to effectively reduce data preparation efforts.

Future work could be devoted to investigating ensembles of J48 trees and applying the proposed algorithm to a corresponding database, where all types of supported attributes naturally arise. Also, a future research direction could be to extend the model to deal with temporal logic formulas. Such a formulation would allow the decision tree to take into account relationships between the values of different attributes, instead of looking at each of them individually.

The motivation is that with the increasing development of cloud computing and big data, traditional sequential decision tree algorithms are no longer suitable.

To evaluate the effectiveness of the presented method, experimental results are shown on a synthetic massive dataset. The empirical results indicate that the implementation of the C4.5 algorithm shows both time efficiency and scalability. In future work, other typical data mining and machine learning algorithms can be investigated using MapReduce.

References

1. Anand, M. (2019). Analytics: Fueling The Digital Economy. Digitalist magazine. Retrieved from https://www.digitalistmag.com/future-of-work/2017/02/06/analytics-fueling-digital-economy- pp. 45-60.

2. Aalst, W.M.P., Pesic, M., & Song, M. (2019). Beyond process mining: From the past to present and future. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 38-52. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-13094-6_5

3. Brynjolfsson, E., & Kahin, B. (2000). Introduction, in Understanding the Digital Economy. / E. Brynjolfsson, B. Kahin (eds). Cambridge: MIT Press. pp. 1-10.

4. Bowyer, J., Hal,l L., Moore, T., Chawla, N., and Kegelmeyer, W. (2000). A parallel decision tree builder for mining very large visualization datasets. Proceedings from MIIM ‘2000: IEEE International Conference *«Systems, Man, and Cybernetics»*, vol. 3, pp. 188-189.

5. Bukht, R., & Hicks, R. (2018). Definition, concept and measurement of the digital economy. Bulletin of international organizations. V. 13, № 2, pp. 143-172. https://doi.org/10.17323/1996-7845-2018-02-07

6. Baruque, B., & Corchado, E. (2019). Fusion methods for unsupervised learning, ensembles, (Vol. 322). Berlin, Germany: Springer. P. 87.

7. Conforti, R., de Leoni, M., La Rosa, M., & van der Aalst, W.M.P. (2013). Supporting risk-informed decisions during business process execution. In: Salinesi, C., Norrie, M.C., Pastor, O. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 116-132. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-38709-8_8

8. Dahlman, C., Mealy, S., Wermelinger, M. (2016). Harnessing the Digital Economy for Developing Countries. Paris: OECD. Retrieved from http://www.oecdilibrary.org/docserver/download/4adffb24-en.pdf

9. Donadello, I., Francescomarino, Ch., Magg, F., Ricci, F., & Shikhizada, A. (2023). Outcome-Oriented Prescriptive Process Monitoring based on Temporal Logic Patterns. Engineering Applications of Artificial Intelligence, Volume 126, Part B, Elsevier, pp. 28-35. https://doi.org/10.1016/j.engappai.2023.106899

10. Data Age 2025. The Digitization of the World. Seagate. Retrieved from https://www.seagate.com/ru/ru/our-story/data-age-2025/

11. Digital Globalization: The New Era of Global Flows. New York, NY: McKinsey Global Institute. Retrieved from mode:https://www.mckinsey.de/sites/mck_files/files/mgi_digital_globalization.pdf;

12. Fournier,-Viger, P., Gomariz, A., Šebek, M., & Hlosta, M. (2014). VGEN: Fast Vertical Mining of Sequential Generator Patterns. In Proceedings from MIIM ’14: Sixteenth International Conference *«Data Warehousing and Knowledge Discovery»*, Munich, Germany, (1-5 September, 2014). (pp. 476-488).

13. Geissbauer, R., Vedso, J., & Schrauf, S. (2016). Industry 4.0: Building the Digital Enterprise. London: Pwc. Retrieved from https://www.pwc.com/gx/en/industries/industries-4.0/landing-page/industry-4.0-building-yourdigital-enterprise-april-2016.pdf

14. Howe, J., Costanzo, M., Fey, P., Gojobor, T.i, Hannic, L.k, & Rhee, S. (2008). Big data: The future of biocuration. Nature, vol. 455, no.7209, pp. 47-50.

15. Kang, B., Kim, D., & Kang, S.H. (2012). Real-time business process monitoring method for prediction of abnormal termination using knni-based lof prediction. Expert Syst. P.120.

16. Lo, D., Cheng, H. (2011). Lucia: Mining closed discriminative dyadic sequential patterns. In: Proc. of EDBT, (pp. 21-32). Springer. https://doi.org/10.1145/1951365.1951371

17. Manyika, J. et al. (2016). Digital Globalization: The New Era of Global Flows. New York, NY: McKinsey Global Institute. Retrieved from https://www.mckinsey.de/sites/mck_files/files/mgi_digital_globalization. pdf (pp. 41-47).

18. OECD (2016). Measuring GDP in a Digitalised Economy. Paris. Retrieved from www.oecd.org/dev/Measuring- GDP-in-a-digitalised-economy.pdf

19. Pesic, M., Schonenberg, H., van der Aalst, W.M.P. (2019). Declare: Full support for loosely structured processes. Proceedings from MIIM ‘19: EDOC, pp. 287-300.

20. Panigrahi, R., & Borah, S. **(**2018). Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets. Procedia Computer Science**,** Volume 132, pp. 323-332.

21. Pika, A., van der Aalst, W.M.P., Fidge, C.J., ter Hofstede, A.H.M., & Wynn, M.T. (2013). Predicting deadline transgressions using event logs. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 211-216. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-36285-9_22

22. Quinlan, J. (1993). C4.5: Programs for Machine Learning. M. Kaufmann Publishers Inc, p. 37.

23. Quinlan, J., (1996). Improved use of continuous attributes in C4.5. ArXiv preprint cs/9603103, pp. 78-82. https://doi.org/10.1613/jair.279

24. Suriadi, S., Ouyang, C., van der Aalst, W.M.P., & der Hofstede, A.H.M. (2013). Root cause analysis with enriched process logs. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops. Lecture Notes in Business Information Processing, vol. 132, pp. 174-186. Springer, Heidelberg. https://doi.org/10.1007/978-3-642-36285-9_18

25. Westergaard. M., & Maggi, F.M. (2011). Modeling and verification of a protocol for operational support using coloured petri nets. In: Kristensen, L.M., Petrucci, L. (eds.) PETRI NETS 2011. LNCS, vol. 6709, pp. 169-188. Springer, Heidelberg.

26. Wolf, F., Brunk J., Becker J. (2021). A Framework of Business Process Monitoring and Prediction Techniques, Springer International Publishing, pp. 43-47. https://doi.org/10.1007/978-3-030-86797-3_47

27. Wil M.P. van der Aalst (2022). Process Mining: A 360 Degree Overview**,** Springer. https://doi.org/10.1007/978-3-031-08848-3_1

28. Xu,M, wang, J., & Chen, T. (2006). Improved decision tree algorithm: ID3+, intelligent computing in signal. Processing and pattern recognition, Vol. 345, pp. 141-149.

ПРОГНОСТИЧНИЙ МОНІТОРИНГ БІЗНЕС-ПРОЦЕСІВ У РАМКАХ РОЗВИТКУ ЦИФРОВОЇ ЕКОНОМІКИ

Т. І. Мшвідобадзе, професор, Горійський державний університет (Грузія)

Методологія дослідження. У цій статті ми представляємо підхід до аналізу таких журналів подій з метою прогнозного моніторингу бізнес-обмежень під час виконання бізнес-процесу. У будь-який момент виконання процесу користувач може визначити бізнес-обмеження у вигляді лінійних правил часової логіки. Фреймворк визначає значення вхідних даних, які з більшою чи меншою ймовірністю відповідають кожному бізнес-обмеженню. Цей підхід був реалізований в інструментах інтелектуального аналізу процесів ProM.

Результати. Результати показують, що ця модель може забезпечити конкурентоспроможну продуктивність класифікації, створювати моделі з високою інтерпретацією та ефективно скоротити зусилля на підготовку даних. Розрахунок критеріїв коефіцієнта інформаційного посилення виконується та відображається в алгоритмі з використанням відповідних рівнянь та рекурсивної функції. У цій статті ми описуємо J48SS, новий індуктор дерева рішень, заснований на алгоритмі C4.5. Алгоритм експериментально перевірено в реальному середовищі аналітики бізнес-мовлення. Показано, що J48SS ефективно зменшує зусилля на підготовку даних, а використання бінарних розбиття показує, що дерево повністю зростає. Подальша робота може бути присвячена дослідженню ансамблів дерев J48 та застосуванню запропонованого алгоритму до відповідної бази даних, де природним чином виникають усі

типи підтримуваних атрибутів. Також, майбутнім напрямком досліджень може бути розширення моделі для роботи з формулами часової логіки. Таке формулювання дозволило б дереву рішень враховувати зв'язки між значеннями різних атрибутів, замість того, щоб розглядати кожен з них окремо.

Новизна. У цій статті представлено J48SS, новий навчальний засіб дерева рішень, який може поєднувати статичні, послідовні дані та дані часових рядів для цілей класифікації. Новий алгоритм базується на популярному навчальному додатку дерева рішень C4.5 та спирається на концепції частого вилучення шаблонів та генерації формул часових рядів. Показано фреймворк для прогнозного моніторингу бізнес-процесів.

Практична значущість. У цій статті ми пропонуємо реалізацію алгоритму MapReduce C4.5. Емпіричні результати показують, що реалізація алгоритму демонструє як ефективність з точки зору часу, так і масштабованість.

Ключові слова: цифрова економіка, моделювання бізнес-процесів, моніторинг процесів, алгоритм C4.5.

Надійшла до редакції 04.05.25 р.