2019 IEEE International Conference on Services Computing (SCC)

Leveraging Shallow Machine Learning to Predict Business Process Behavior

Annalisa Appice, Nicola Di Mauro, Donato Malerba
Department of Computer Science
University of Bari Aldo Moro
Bari, Italy
Email: [annalisa.appice,nicola.dimauro,donato.malerba]@uniba.it

***Abstract*—This study investigates facets of shallow machine learning as an accurate data-centric approach to predict business process behaviour. Shallow machine learning is investigated as a part of a holistic approach that combines feature construction, local and global learning, classification and regression algorithms. Experiments show that, despite the emerging attention towards deep learning also in predictive process mining, stacking feature construction and shallow machine learning algorithms can still outperform various process predictor competitors (included deep learning ones).**

***Keywords*-process prediction, feature construction, machine learning**

I. Introduction

Being able to predict the future behaviour of a business process is an important business capability that can guarantee the higher utilization by acting proactively in anticipation. As an application of predictive analytics, process prediction is mainly concerned with predicting the evolution of running traces based on models extracted from a historical event log [1]. Examples include techniques to predict the completion cycle time until a trace is resolved [2], [3], [4], [5], the next activity [6], [7], [8], [9], [4], [10] and the timestamp of the activity [11], [4], [2], [7], as well as the outcome of a trace [12], [13].

General purpose predictive analytics has a long history in supervised shallow machine learning and, more recently, also in deep learning. Shallow learning requires a feature engineer to perform the task of identifying the relevant data characteristics before executing well known classification and regression algorithms. On the other hand, deep learning relies on a complex multi-layered representation of the input data and can perform feature engineering autonomously through a process defined representation learning. As deep learning is gaining momentum, it is attracted attention also in predictive process mining [8], [4], [14], where deep architectures have been defined to learn over sequence of events. However, in this paper, we would investigate how shallow learning can still provide robust predictive analytics in the context of process mining.

A process prediction task can be in principle addressed into a shallow machine learning context after that a training set of data is extracted from a historical event log to fuel supervised classification or regression algorithms. This training set should comprise a feature space with the target to predict. Then a classification or regression model can be learned from this training set so that the learned model will permit one to predict the corresponding target of a new example over a running trace based solely on its descriptive features. Motivated by the previous considerations, the main contribution of this paper is a holistic data-centric approach that applies shallow classification and regression algorithms for predictive modeling of business processes.

The effectiveness of the proposed approach is empirically evaluated in two benchmark event logs. The evaluation assesses the viability of the constructed feature space along the predictive process tasks and the machine learning algorithms. It also investigates the accuracy dichotomy in choosing whether to use local versus global modeling with respect to the predictive task. Finally, it compares the accuracy of the proposed approach to that of recent competitors also designed in the emerging field of deep learning. Interestingly, our investigations reveal that competitive predictive accuracy can be still achieved by a shallow learning approach with the result of outperforming complex almost unfathomable neural network based predictions.

The paper is organized as follows. Section II reports preliminary concepts and Section III illustrates the proposed process machine learning approach. Section IV describes the relevant results of the empirical study. Finally, Section V draws some conclusions and outlines some future work.

II. Preliminary concepts

The basic assumption is that an event log contains events that register information on activities executed for specific traces of a certain process type, as well as their duration. The activity names belong to a finite, non empty set, denoted as $\mathcal{A}$. An *event* ϵ is characterized by two mandatory characteristics, that is, the event contains an activity name A (with $A \in \mathcal{A}$) and has a timestamp t representing date and time of occurrence. An event log is a set of events, where each event is linked to a particular trace. A *trace* $\mathcal{T}$ represents the execution of a business process instance. It is a finite (non empty) sequence of distinct events such that time is non-decreasing in the trace—i.e., $t_i \leq t_j$, $1 \leq i < j$. The length of a trace is the number of events in the trace. An *event log* $\mathcal{L}$ is a bag of traces. A *prefix trace* is a sub-sequence of a

2474-2473/19/$31.00 ©2019 IEEE
DOI 10.1109/SCC.2019.00039

trace starting from the beginning of the trace. Formally, let $\mathcal{T}$ be a trace made up of a sequence of l events, a prefix trace $\mathcal{PT}_{1...k}$ is the sequence of the first k consecutive events of $\mathcal{T}$ with $1 \leq k < l$. No prefix trace can correspond to the original trace (as $k < l$ in the definition). A prefix trace can be represented over a descriptive feature space—i.e., a vector of features. Every *feature* corresponds to a metric that assigns a value to each prefix trace. Thus, embedding an event log to a vector representation corresponds to defining a function assigning each prefix trace of the event log to a vector of values—one value for each feature. The resulting embeddings with the associated targets can be used as a *training set* to learn a predictive model allowing to infer the target of new examples over a running trace. Let $\mathcal{PT}_{1...k}$ be a prefix trace of a trace $\mathcal{T}$, then the *target* for the *next activity* task is A_{k+1}—the activity corresponding to the event $\epsilon_{k+1} \in \mathcal{T}$—while the *target* for the *next activity* time is $(t_{k+1} - t_k)$—the time elapsed between events ϵ_k and ϵ_{k+1} of $\mathcal{T}$. Finally, the *target* for the *cycle time* is $(t_l - t_k)$—the time elapsed between the last trace event ϵ_l and ϵ_k.

III. Machine learning predictive approach

For each target variable (next activity or timestamp of the next activity or completion cycle time) considered in this study, the proposed approach transforms a historical event log into a training dataset (Section III-A). It runs shallow machine learning algorithms, in order to learn process prediction models from the training data (Section III-B). Finally, it uses these models to predict the process behaviour of a new running trace (Section III-C).

A. Extracting descriptive features

Prefix traces of a historical event log are transformed into feature space vectors populated with descriptive features related to the control-flow perspective (order of activities), the trace perspective (frequency of activities) and the performance perspective (time performance). Two feature extraction approaches, namely ALL and WINDOW, are considered.

In the ALL approach, the entire prefix traces are used to populate the descriptive feature space. Specifically, for any combination of two activity names, e.g. (A, B) in the activity domain $\mathcal{A}$, a transition feature is defined ($A \rightarrow B$). This counts how many times an event with the activity named A has been directly followed by another event with the activity named B in the prefix trace. For each activity name in $A \in \mathcal{A}$, one activity feature is constructed. This is measured by counting the number of events of the prefix trace having the specified activity's name. Finally, a predefined set of performance features is considered, in order to represent the temporal information. Performance features are: the length and time duration of the prefix trace, as well as the minimum, maximum, mean and median time difference computed between consecutive events in the prefix trace.

In the WINDOW approach, the most recent w events, which form the w-long suffix of the prefix traces, are selected. Windowed events are used to populate the descriptive feature space. In particular, w activity features are defined by considering the activity information. Each feature represents the name of the activity in the windowed event. As these features describe the name of the activities executed at w consecutive time points, they also describe activity transitions over the window. In addition, $w-1$ performance features are defined by considering the timestamp. Each feature represents the time difference computed between consecutive windowed events. Whenever the size of the prefix trace is lower than w, a dummy event represents each unavailable event. The dummy event has the activity name equal to $none$ and the timestamp equal to the starting time point of the prefix trace.

B. Learning process prediction models

Any vector-based classification algorithm can be selected to predict the next activity, while any regression algorithm can be selected to predict the timestamp of the next activity or the completion cycle time of the trace. In addition, both global and local learning approaches can be applied in combination with both classification and regression algorithms. In the global learning approach, for each target variable, one predictive model is learned for the entire training dataset. On the other hand, in the local learning approach, for each target variable, a vector of predictive models is narrowed down to the prefix traces, which share the same latest activity. In this way, a local prediction model can be learned for each distinct activity type that is the last one executed in the prefix trace. A prefix trace ending with activity A_i belongs to the i-th local training set. For each target variable, a local predictive model is learned from each local training set.

C. Predicting process behavior

Let us consider a running trace. This is dealt as a prefix trace with unknown targets. It is transformed into the descriptive feature vector so that the process behavior models, trained from a historical event log, can be used to predict the activity and the timestamp of the next event, as well as the completion time. If predictive models have been learned using the local approach, prediction models one-to-one associated to the latest activity executed in the running trace are selected to yield the predictions.

IV. Experiments

Let pmKOMETA denote the predictive approach described in this paper. It is evaluated in the three predictive tasks at hand by considering the benchmark event logs and the experimental setting described in [4].

A. Event logs and experimental setting

Helpdesk log[1] contains events from a ticketing management process of the help desk of an Italian software company. The business process consists of 9 activities. This log contains 3804 traces and 13710 events. BPI challenge 2012 event log[2] pertains to an application process for a personal loan or overdraft within a global financing organization. Based upon considerations discussed in [4], the evaluation is narrowed down to the 9658 work item traces, which contains events that are manually executed. In addition, only 72414 events of 6 activities with type complete are retained. This pre-processing is performed as in [4].

For each event log, the chronologically ordered first 2/3 of the traces are used as training event log, while the accuracy of predictions is evaluated on the remaining 1/3 of the traces. The next activity and its timestamp predictions, as well as the completion cycle time are evaluated on all testing prefix traces. As reported in [4] no prediction is performed for 1-sized prefix traces.

Two configurations are compared along the feature construction scheme (see Section III-A): ALL and WINDOW. Based on preliminary investigations, the best accuracy with WINDOW is commonly achieved with window length set equal to 6. Two configurations are also compared along the machine learning approach (see Section III-B): G, the global learning approach and L, the local learning approach. Finally, four configurations are considered along the choice of the base classification algorithm: i) kNN—k-Nearest Neighborhood (with Euclidean Distance, number of nearest neighbors equal to 50 and weighting mechanism with the inverse of the distance), ii) RF—Random Forest (with number of trees equal to 50), iii) J48—decision tree, and iv) LOG—Logistic Regression (with logistic parameter equal ranging among $1.0e-12, 1.0e-11, \ldots,$ and $1.0e+2$ and automatically selected with a grid search on a three-fold cross validation of the training set). Three configurations are defined along the choice of the base regression algorithm: kNN, M5'—model tree, and SVR—Support Vector Regression (with Gaussian kernel, C=64 and γ parameter ranging among $2.0E-3, 2.0E-2, \ldots$ and $2.0E+3$ and automatically selected with a grid search on a 3-fold cross validation of the training set)[3]. These algorithms are selected as commonly used in many machine learning applications [15].

The experimental setting is replicated to evaluate the performance of various configurations of pmKOMETA and its competitors. We consider four competitors for the prediction of the next activity [7], [9], [8], [4], three competitors for the prediction of the timestamp [2], [7], [4] and three competitors for the prediction of the completion cycle time [3], [2], [4]. Results of competitors are collected from [4].

[1]https://data.mendeley.com/datasets/39bp3vv62t/1.

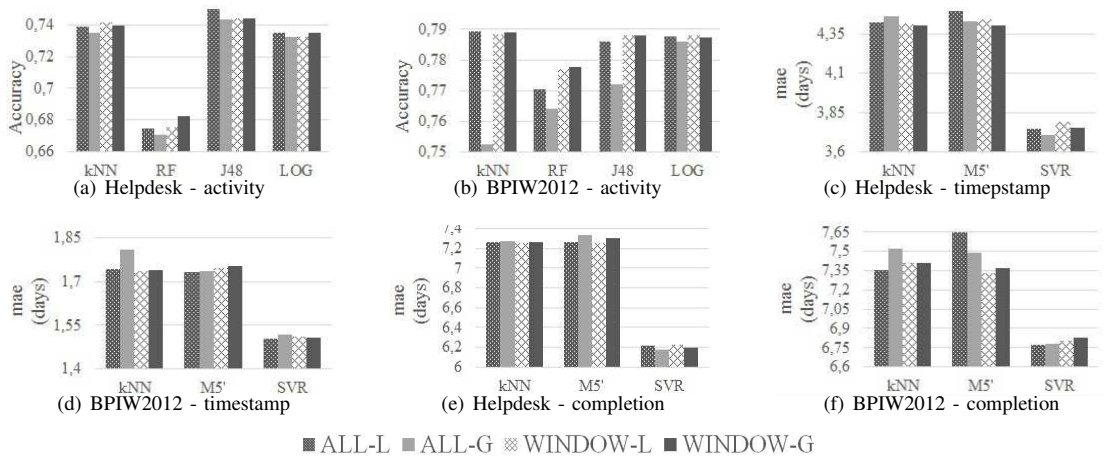
[2]http://www.win.tue.nl/bpi/2012/challenge

[3]The learning algorithms implemented in WEKA 3.6 are used.

B. Results and discussion

We start analyzing pmKOMETA in the prediction of the next activity. The accuracy results in Figures 1(a)-1(b) confirm that local approach L commonly yields a gain in the accuracy for this specific task. Interestingly, in both event logs, configurations with L always outperform configurations with G when the descriptive features are constructed in configuration ALL. This behavior is independent of the base classification algorithm. To improve this analysis, we explore the accuracy achieved by the most accurate configurations of pmKOMETA, which are selected for both L (row 1, Table I – L+ALL+J48 for Helpdesk and L+ALL+kNN for BPIW2012) and G (row 2, Table I, G+ALL+kNN for Helpdesk and G+WINDOW+J48 for BPIW2012), respectively. The highest accuracy is achieved with L+ALL in both event logs, although the classification algorithm may change in the selected configurations. This confirms that the combination of local approach L and features computed with schema ALL actually contributes to yield accurate predictions of the next activity by overcoming the difficulty posed by predicting the behaviour of traces executing complex process models. On the other hand, these results do not provide sufficient evidence to assess that a classification algorithm can systematically outperform all other classification algorithms in this task. Both kNN and J48 are alternately selected in the best configurations for the two event logs. Final considerations are drawn from the analysis of competitors (rows 3-6, Table I). pmKOMETA gains in accuracy compared to all competitors in this study included the deep learning competitors proposed in [8], [4].

We proceed analyzing pmKOMETA in the prediction of the timestamp of the next activity and the completion cycle time. The error results, reported in Figures 1(c)-1(d) for the prediction of the timestamp of the next activity and Figures 1(e)-1(f) for the prediction of the completion cycle time, highlight that local approach L not necessarily diminishes the error. Instead, results allow us to conclude that the regression algorithm SVR always leads to a reduction of the error and that this behavior is independent of the modeling approach and the feature schema selected. Finally, configurations with ALL generally outperform configurations with WINDOW. These considerations are also confirmed by the best configurations of pmKOMETA, which are selected for both L (rows 7 and 11, Table I – L+ALL+SVR for both tasks, as well as the two logs) and G (rows 8 and 13, Table I, G+ALL+SVR for the two tasks in Helpdesk, G+WINDOW+SVR for the prediction of the timestamp of the next activity in BPIW2012, while G+ALL+SVR for the prediction of the completion cycle time in BPIW2012) respectively. Finally, we note that for both event logs the best configurations selected for pmKOMETA always outperform the best configuration of each competitor in this study (rows



(a) Helpdesk - activity (b) BPIW2012 - activity (c) Helpdesk - timepstamp

(d) BPIW2012 - timestamp (e) Helpdesk - completion (f) BPIW2012 - completion

Figure 1. Accuracy of next activity predictions (Figures 1(a) and 1(b)), mean absolute error (in days) of timestamp of the next activity predictions (Figures 1(c) and 1(d)) and mean absolute error (in days) of completion cycle time predictions (Figures 1(e) and 1(f)) of various configurations of pmKOMETA.

Table I
ACCURACY OF NEXT ACTIVITY PREDICTIONS (ROWS 1-6), MEAN ABSOLUTE ERROR (IN DAYS) OF TIMESTAMP OF NEXT ACTIVITY PREDICTIONS (ROWS 7-11) AND MEAN ABSOLUTE ERROR (IN DAYS) OF COMPLETION CYCLE TIME PREDICTIONS (ROWS 12-13). THE REPORTED CONFIGURATIONS OF pmKOMETA (ROWS 1-2, 7-8, 12-13) ARE THE MOST ACCURATE CONFIGURATIONS SELECTED WITH RESPECT TO THE LOCAL MODELING APPROACH AND THE GLOBAL MODELING APPROACH. THE BEST RESULTS ARE IN BOLD. RESULTS OF COMPETITORS FOR COMPLETION TIME PREDICTION ARE REPORTED IN FIGURE 2 GROUPED FOR PREFIX TRACES OF DIFFERENT LENGTH AS REPORTED IN [4].

task	Helpdesk		BPIW2012	
activity	L+ALL+J48	**.753**	L+ALL+kNN	**.789**
	G+ALL+kNN	.739	G+WINDOW+J48	.787
	[7]	.732	[7]	.785
	[4]	.712	[4]	.760
			[8]	.623
			[9]	.719
timestamp	L+ALL+SVR	3.74	L+ALL+SVR	**1.50**
	G+ALL+SVR	**3.70**	G+WINDOW+SVR	**1.50**
	[7]	4.42	[7]	1.76
	[4]	3.75	[4]	1.56
	[2]	5.67	[2]	1.91
completion	L+ALL+ SVR	6.21	L+ALL+SVR	**6.77**
	G+ALL+ SVR	**6.18**	G+ALL+SVR	**6.77**

9-11, Table I, as well as Figures 2(a)and 2(b)).

C. Final remarks

In short, the empirical study shows that local machine learning approach L yields more accurate predictions than global machine learning approach G only when we use L to predict the next activity. This behavior depends on the fact that L derives a distinct predictive model based on the last activity observed in the training prefix traces. This appears relevant for the activity prediction only. In fact, the prediction of the next activity is reasonably dependent on the activities already performed. Differently, this information is not so relevant for the time predictions. Additional remarks can be drawn from the overall analysis of both the feature construction and the predictive algorithms. The features constructed considering all events of the prefix trace commonly contribute to achieve to the highest accuracy in each task. Both Decision Trees and k-Nearest Neighborhood perform well for the next activity prediction, while Support Vector Regression achieves the lowest error for the prediction of both the timestamp of the next activity and the completion cycle time. Finally the empirical study highlights that pmKOMETA outperforms existing competitors, included deep learning ones. This is a further confirmation of the viability of the shallow machine learning holisitc approach that is the main contribution of this paper. A limitation of the proposed approach is that the proposed learning approach does not include any mechanism to incorporate possible interventions, which, in turn, may be operated by resources running the traces once they know predictions of the trace future behavior during the execution. In addition, the learning model is performed in a batch mode only by considering historical data, without any ability of fitting the predictive model to possible changes occurring in the process behavior over the time.

V. CONCLUSION AND FUTURE WORK

This paper describes an integrated machine learning approach, in order to yield accurate predictions of the next activity in a running trace, the timestamp of the next activity, as well as the completion cycle time. The empirical study shows the effectiveness of the proposed approach compared to various competitors also developed in deep learning.

As future work, we plan to investigate new clustering solution based on time performance to improve the performance of local machine learning in predicting the time performance behavior of a trace. We also intend to extend the proposed approach to stream learning, in order to learn predictive

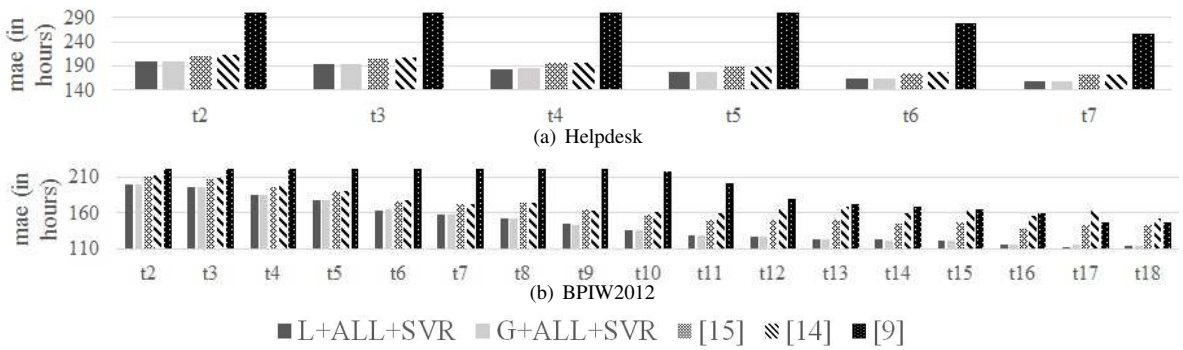


Figure 2. Mean absolute error (in hours) of completion cycle time prediction using prefix traces of different length. The errors (in hours) of Dongen et al. 2008 [3], van der Aalst et al. 2011 [2] and Tax eta al. 2017 [4] are provided by Niek Tax and reported in [4].

models which may also change over the time as new traces are collected. We also plan to explore the use of prescriptive learning theories, in order to enrich the proposed learning approach with guidelines that describe what to do in order to achieve specific outcomes and how expanding the model by integrating possible reactions to prediction-based alerts.

Acknowledgments: The research is partially supported by the POR Puglia FESR-FSE 2014-2020 - Asse prioritario 1 - Ricerca, sviluppo tecnologico, innovazione - Sub Azione 1.4.b Bando INNOLABS - Sostegno alla creazione di soluzioni innovative finalizzate a specifici problemi di rilevanza sociale - *Research project KOMETA (Knowledge Community for Efficient Training through Virtual Technologies)*, funded by Regione Puglia. The authors wish to thank Niek Tax for providing the detailed results of the research in [4], as well as ReCaS-Bari resource team for providing the infrastructure to run the experimental study.

REFERENCES

[1] C. Di Francescomarino, C. Ghidini, F. M. Maggi, and F. Milani, "Predictive process monitoring methods: Which one suits me best?" in *Business Process Management*. Springer, 2018, pp. 462–479.

[2] W. M. P. van der Aalst, M. H. Schonenberg, and M. Song, "Time prediction based on process mining," *Information Systems*, vol. 36, no. 2, pp. 450–475, 2011.

[3] B. F. van Dongen, R. A. Crooy, and W. M. P. van der Aalst, "Cycle time prediction: When will this case finally be finished?" in *On the Move to Meaningful Internet Systems*. Springer, 2008, pp. 319–336.

[4] N. Tax, I. Verenich, M. La Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in *International Conference on Advanced Information Systems Engineering*. Springer, 2017, pp. 477–492.

[5] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, and I. Teinemaa, "Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring," *CoRR*, vol. abs/1805.02896, 2018.

[6] S. Pravilovic, A. Appice, and D. Malerba, "Process mining to forecast the future of running cases," in *New Frontiers in Mining Complex Patterns*, ser. LNCS, vol. 8399. Springer, 2014, pp. 67–81.

[7] A. Appice, D. Malerba, V. Morreale, and G. Vella, "Business event forecasting," in *10th International Forum on Knowledge Asset Dynamics*, 2015, pp. 1442–1453.

[8] J. Evermann, J.-R. Rehse, and P. Fettke, "A deep learning approach for predicting process behaviour at runtime," in *Business Process Management Workshops*. Springer, 2017, pp. 327–338.

[9] D. Breuker, M. Matzner, P. Delfmann, and J. Becker, "Comprehensible predictive models for business processes," *Journal MIS Quarterly*, vol. 40, pp. 1009–1034, 2016.

[10] M. Le, B. Gabrys, and D. Nauck, "A hybrid model for business process event prediction," in *Research and Development in Intelligent Systems*. Springer, 2012, pp. 179–192.

[11] K. Böhmer and S. Rinderle-Ma, "Probability based heuristic for predictive business process monitoring," in *On the Move to Meaningful Internet Systems*. Springer, 2018, pp. 78–96.

[12] I. Teinemaa, M. Dumas, A. Leontjeva, and F. M. Maggi, "Temporal stability in predictive process monitoring," *DMKD*, vol. 32, no. 5, pp. 1306–1338, 2018.

[13] C. D. Francescomarino, M. Dumas, M. Federici, C. Ghidini, F. M. Maggi, W. Rizzi, and L. Simonetto, "Genetic algorithms for hyperparameter optimization in predictive business process monitoring," *Information Systems*, vol. 74, pp. 67 – 83, 2018.

[14] C. Di Francescomarino, C. Ghidini, F. M. Maggi, G. Petrucci, and A. Yeshchenko, "An eye into the future: Leveraging a-priori knowledge in predictive business process monitoring," in *Business Process Management*. Springer, 2017, pp. 252–268.

[15] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, 2008.