



Proactive Process Adaptation Using Deep Learning Ensembles

Andreas Metzger^(✉) , Adrian Neubauer, Philipp Bohn, and Klaus Pohl

paluno – The Ruhr Institute for Software Technology,
University of Duisburg-Essen, Essen, Germany
{andreas.metzger, adrian.neubauer, philipp.bohn,
klaus.pohl}@paluno.uni-due.de

Abstract. Proactive process adaptation can prevent and mitigate upcoming problems during process execution. Proactive adaptation decisions are based on predictions about how an ongoing process instance will unfold up to its completion. On the one hand, these predictions must have high accuracy, as, for instance, false negative predictions mean that necessary adaptations are missed. On the other hand, these predictions should be produced early during process execution, as this leaves more time for adaptations, which typically have non-negligible latencies. However, there is an important tradeoff between prediction accuracy and earliness. Later predictions typically have a higher accuracy, because more information about the ongoing process instance is available. To address this tradeoff, we use an ensemble of deep learning models that can produce predictions at arbitrary points during process execution and that provides reliability estimates for each prediction. We use these reliability estimates to dynamically determine the earliest prediction with sufficient accuracy, which is used as basis for proactive adaptation. Experimental results indicate that our dynamic approach may offer cost savings of 27% on average when compared to using a static prediction point.

Keywords: Business process monitoring · Proactive adaptation · Prediction · Accuracy · Earliness

1 Introduction

Proactive process adaptation can prevent the occurrence of problems and it can mitigate the impact of upcoming problems during process execution [1, 18, 31] by dynamically re-planning the flow of a running process instance [19, 28, 37]. Proactive process adaptation thereby can avoid contractual penalties or time-consuming roll-back and compensation activities.

Proactive process adaptation relies on predictive process monitoring to predict potential problems. Predictive process monitoring predicts how an ongoing process instance (a.k.a. case) will unfold up to its completion [8, 16, 22]. If a potential problem is predicted, adaptation decisions are taken at run time

to prevent or mitigate the predicted problem. As an example, a delay in the expected delivery time for a freight transport process may incur contractual penalties [11]. If during the execution of such freight transport process a delay is predicted, faster transport services (such as air delivery instead of road delivery) can be proactively scheduled in order to prevent the delay.

With respect to predictions, there are two important requirements for proactive process adaptation. On the one hand, predictions must have high accuracy, as, for instance, false negative predictions mean that necessary adaptations are missed. On the other hand, predictions should be produced early during process execution, as this leaves more time for adaptations, which typically have non-negligible latencies. However, there is an important tradeoff between these two requirements. Later predictions typically have a higher accuracy, because more information about the ongoing process instance becomes available.

We address the aforementioned tradeoff by using ensembles of deep learning models. Ensemble prediction is a meta-prediction technique where the predictions of m prediction models are combined into a single prediction [30]. We use deep learning ensembles to produce predictions at arbitrary points during process execution. In addition we compute reliability estimates for each prediction by computing the fraction of prediction models that predicted the majority class [19]. A high reliability indicates a high probability that the ensemble prediction is correct. We use these reliability estimates to dynamically determine the earliest prediction with sufficiently high reliability and use this prediction as basis for proactive adaptation. Experimental results based on four real-world data sets suggests that our dynamic approach offers cost savings of 27% on average when compared to using a fixed, static prediction point.

Section 2 provides a detailed problem statement and analysis of related work. Section 3 describes our approach. Section 4 provides its experimental evaluation.

2 Problem Statement and Related Work

2.1 Prediction Accuracy and Reliability

Problem. As mentioned above, one key requirement for proactive process adaptation are accurate predictions. Informally, prediction accuracy characterizes the ability of a prediction technique to forecast as many true violations as possible, while generating as few false alarms as possible [33]. Prediction accuracy is important for two main reasons [21]. First, accurate predictions deliver more true violations and thus trigger more required adaptations. Each missed required adaptation means one less opportunity for preventing or mitigating a problem. Second, accurate predictions mean less false alarms, and thus triggering less unnecessary adaptations. Unnecessary adaptations incur additional costs for executing the adaptations, while not addressing actual problems.

Previous research on predictive process monitoring (see [16] for an overview) focused on aggregate accuracy, such as precision or recall. Even though a high aggregate accuracy is beneficial, it does not provide direct information about the accuracy of an *individual* prediction. Knowing the accuracy of an individual

prediction is important, because some predictions may have a higher probability of being correct than others. Proactive adaptation decisions are taken on a case by case basis. Therefore, the information about whether an individual prediction may be correct provides additional support for decision making [18, 19].

Prediction techniques traditionally used for predictive process monitoring (such as decision trees, k-nearest-neighbors, support vector machines, and multi-layer perceptrons [8, 16]) can provide probabilities to indicate whether an individual prediction is correct (e.g., in the form of class probabilities of a decision tree). Yet, probabilities estimated by most of these prediction techniques are poor [38]. In contrast, so called *reliability estimates* (e.g., computed from ensembles of prediction models) can provide better estimates of the probability that an individual prediction is correct [2].

Related Work. To improve aggregate prediction accuracy, deep learning techniques are being employed for predictive process monitoring [5, 9, 17, 26, 27, 34]. In particular, Recurrent Neural Networks (RNNs) are employed, which are a special type of artificial neural network, where each neuron also feeds back information into itself [5, 10]. Empirical evidence indicates that RNNs provide significant accuracy improvements for predictive process monitoring [5, 20, 34]. As an example, the empirical results of our previous work on using RNNs show an accuracy improvement of 36% when compared to multi-layer perceptrons [20]. Yet, the aforementioned approaches only consider aggregate accuracy and do not consider the accuracy of individual predictions for proactive adaptation decisions.

In the literature, some authors considered the probability that an individual prediction is correct. Maggi et al. [15] use decision tree learning for predictive process monitoring. As a follow up, Francescomarino et al. [7] employ random forests (an ensembles of decision trees) for prediction. Both use class probabilities of decision trees. They analyze how selecting predictions using class probabilities impacts on aggregate prediction accuracy. They observe that using class probabilities may improve aggregate accuracy, but at the expedient of losing predictions that are below a given probability threshold. Yet, they do not analyze in how far using these class probabilities may improve proactive process adaptation and whether it may offer cost savings.

In our earlier work, we used reliability estimates computed from ensembles of multi-layer perceptrons to decide on proactive adaptation [18, 19]. If the reliability for a given prediction is equal to or greater than a predefined threshold, the prediction is used to trigger a proactive adaptation. In [19] we considered reliabilities computed from ensembles of classification models, which led to cost savings of up to 54% (14% on average). In [18] we also included the magnitude of a predicted violation (computed from ensembles of regression models) into the adaptation decision, which led to additional cost savings of up to 31% (14.8% on average). Yet, we used a fixed point for our predictions (the 50% mark of process execution), and thus did not consider the aspect of prediction earliness.

2.2 Prediction Earliness

Problem. Predictions can be made at different points during the execution of a process instance. The point during process execution for which a prediction is made is called *checkpoint* [13,22]. When determining checkpoints, there is an important tradeoff to be taken into account between prediction accuracy and the earliness of the prediction [13]. This is particularly important when predictions at a given checkpoint are used as basis for proactive process adaptation.

Typically, prediction accuracy increases as the process unfolds, as more information about the process instance becomes available. As an example, between the 25% mark and the 75% mark in process execution, accuracy may increase by 44% (as reported in [36]) and even by 97% (as reported in [22]). This means later predictions have a higher chance to be correct predictions, and thus one should favor later checkpoints as basis for proactive process adaptation.

However, waiting for the predictions of later checkpoints also means that the remaining time for proactively addressing problems becomes shorter [13]. This can be important as adaptations typically have non-negligible latencies, i.e., it may take some time until they become effective [23]. As an example, dispatching additional personnel to mitigate delays in container transports may take several hours. Also, the later a process is adapted, the fewer options may be available for adaptation. As an example, while at the beginning of a transport process one may be able to transport a container by train instead of ship, once the container is on-board the ship, such adaptation may no longer be feasible. Finally, if an adaptation is performed late in the process and turns out not to be effective, not much time may remain for any remedial actions or further adaptations. This means one should choose a rather early checkpoint.

Related Work. In the literature, the tradeoff between prediction accuracy and earliness was approached from different angles. Several authors use prediction earliness as a dependent variable in their experiments. This means they evaluate their proposed predictive process monitoring techniques by considering prediction earliness in addition to prediction accuracy. As an example, Kang et al. [12], Teinemaa et al. [36], and we in our earlier work [20,22] measured the accuracy of different prediction techniques for the different checkpoints along process execution. Results presented in the aforementioned works clearly show the tradeoff between prediction earliness and accuracy. However, how to resolve the tradeoff between accuracy and earliness was not further addressed.

To increase the earliness of accurate predictions, several authors proposed new variants of prediction techniques. As an example, Teinemaa et al. investigate whether unstructured data may increase prediction earliness and accuracy [35]. Similarly, Leontjeva et al. exploit the data payload of process events to increase prediction earliness [14]. Finally, Francescomarino et al. investigate in how far hyper-parameter optimization [7] and clustering [6] can improve earliness. A similar tradeoff between accuracy and earliness was investigated for time series classification, i.e., for predicting the class label of a temporally-indexed set of data points. The aim is to predict the final label of a time series with sufficiently

high accuracy by using the lowest number of data points. Being able to accurately classify a time series early on facilitates early situation detection and thus may help to timely respond to risks and failures [24]. An additional motivation is to reduce the computational effort when compared with using the whole time series for prediction, which is of particular concern for resource- or power-constrained devices [25, 29]. As an example, Mori et al. use probabilistic classifiers to produce a class label for a time series as soon as the probability at a checkpoint exceeds a class-dependent threshold [25]. The aforementioned works address different needs of earliness and accuracy by setting the available parameters, such as prediction reliability thresholds. However, they did not examine in how far the techniques have found a good trade-off between earliness and accuracy. Doing so, requires quantifying the utility of the achieved trade-off, as comparing the techniques solely based on earliness and accuracy may not provide a fair comparison. To quantify such utility, we thus measure how choosing the actual checkpoint for adaptation decisions impacts on overall costs of process execution.

3 Deep Learning Ensembles for Proactive Adaptation

To find a trade-off between earliness and accuracy, we exploit the fact that reliability estimates can provide information about the accuracy of an individual prediction. The key idea of our approach is to (i) dynamically determine, for each process instance, the earliest checkpoint that delivers a sufficiently high reliability, and (ii) use this checkpoint to decide on proactive adaptation.

We compute predictions and reliability estimates from ensembles of deep learning models, specifically RNN models. Ensemble prediction is a meta-prediction technique where the predictions of m prediction models, so called *base learners*, are combined into a single prediction [30].

Ensemble prediction is primarily used to increase aggregate prediction accuracy, while it also allows computing reliability estimates (see Sect. 2.1). Computing reliability estimates is the main reason why we use ensembles of RNN models in our approach. As added benefit, predictions computed via such ensembles provide higher prediction accuracy than using a single RNN model. As an example, RNN ensembles provide an 8.4% higher accuracy when compared with a single RNN model (as used in [20]).

Figure 1 provides an overview of the main activities of our approach.

The ensemble of RNN models creates a prediction T_j at each potential checkpoint j . In addition, it provides the reliability estimate ρ_j for this prediction. If the ensemble predicts a violation at checkpoint j , a proactive adaptation may be needed in order to prevent or mitigate the predicted violation. However, we only act on this prediction if its reliability ρ_j is equal to or greater than a predefined threshold, i.e., we only act if we consider the prediction reliable enough. Thereby, our approach dynamically determines the earliest checkpoint with sufficient reliability that is used as basis for proactive adaptation. This implies that the actual checkpoint chosen for a proactive adaptation decision will vary among the different process instances, in the same way the reliability estimates may be different for each prediction and each process instance.

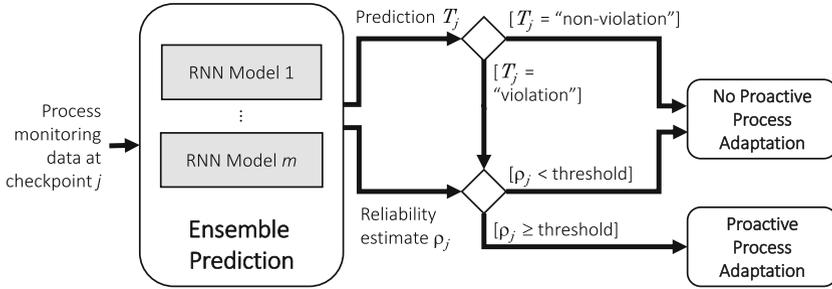


Fig. 1. RNN ensemble for dynamically deciding on proactive process adaptation

3.1 RNNs as Base Learners

We use RNNs as base learners, i.e., as the individual models in the ensemble, as RNNs can handle arbitrary length sequences of input data [10]. Thus, a single RNN can be employed to make predictions for business processes that have an arbitrary length in terms of process activities. In contrast, other prediction techniques (such as random forests or multi-layer perceptrons) either require training a prediction model for each of the checkpoints or they require the special encoding of the input data to train a single model [16, 22, 36]. However, these encodings entail information loss and thus may limit prediction performance.

RNNs also facilitate the scalability of our dynamic approach. Assume we have c checkpoints in the business process. A single RNN model can make predictions at any of these c checkpoints [5, 34]. If we want to avoid information loss, other prediction techniques would require the training of c prediction models, one for each of the c checkpoints. Our exploratory performance measurements indicate a training time of ca. 8 min per checkpoint for multi-layer perceptrons on a standard PC, while the training time for an RNN was 25 min¹. This means that RNNs provide better scalability for our approach if the process has many potential checkpoints ($c > 3$ in our case).

We use RNNs with Long Short-Term Memory (LSTM) cells as they better capture long-term dependencies in the data [17, 34]. Our implementation of these RNN base learners is available online². It exploits the Keras library³ running on top of TensorFlow⁴.

However, RNNs also face specific challenges when used for predictive process monitoring. Even though the data that is fed into an RNN is sequential, i.e., a sequence of events, these events represent the execution of business processes which may include loops and parallel regions. Such non-sequential control flows

¹ Further performance speedups are possible via special-purpose hardware and RNN implementations. RNN training time reduced to 8 min on GPUs (using CuDNN), and further to 2 min on TPUs (Tensor Processing Units).

² <https://github.com/ChemSORly/BusinessProcessOutcomePrediction>.

³ <https://keras.io/>; Version 2.2.4.

⁴ <https://github.com/tensorflow/>; Version 1.9.

can make prediction with RNNs more difficult [5, 9, 34], as RNNs were conceived for natural language processing, which is sequential by nature [10].

To address these difficulties, we employ the following two solutions (presented in earlier work [20]). First, instead of incrementally predicting the next process event until we reach the final event and thus the process outcome (such as proposed in [5, 9, 34]), we directly predict the process outcome. Thereby, we avoid the problem RNNs may have in predicting the next process activity when process execution entails loops with many repeated activities [9, 34]. Second, we encode parallel process activities by embedding the branch information as an additional attribute of the respective process activity. Thereby, we address the problem that parallel process activities can make the prediction task more difficult [5].

3.2 RNN Ensembles

We use bagging (bootstrap aggregating [4]) as a concrete ensemble technique to build the base learners of our RNN ensemble. Bagging generates m new training data sets from the whole training set by sampling from the whole training data set uniformly and with replacement. For each of the m new training data sets an individual RNN model is trained. We use bagging with a sample size of 60% to increase the diversity of the RNN ensembles. Generating the ensembles using bagging also contributes to the scalability of our approach, as the training of the base learners can happen in parallel.

For computing the ensemble predictions T_j and reliability estimates ρ_j for each checkpoint j , we employ the strategies defined in [18, 19], because these strategies showed reasonably good results for a fixed checkpoint. Let us assume that at each checkpoint j , each of the m base learners of the ensemble delivers a prediction result $T_{i,j}$, with $i = 1, \dots, m$, where $T_{i,j}$ is either of class “violation” or “non-violation”.

The **ensemble prediction** for checkpoint j is computed as a majority vote:

$$T_j = \begin{cases} \text{“violation”}, & |i : T_{i,j} = \text{“violation”}| \geq m/2 \\ \text{“non-violation”}, & \text{otherwise.} \end{cases}$$

The **reliability estimate** ρ_j for prediction T_j is computed as the fraction of base learners that predicted the majority class:

$$\rho_j = \max_{i=1, \dots, m} \left(\frac{|i : T_{i,j} = \text{“violation”}|}{m}, \frac{|i : T_{i,j} = \text{“non-violation”}|}{m} \right)$$

4 Experimental Evaluation

4.1 Cost Model

We aim to answer the question in how far determining checkpoints based on reliability estimates (our *dynamic* approach) compares to determining checkpoints

based on aggregate accuracy (the *static* approach). To this end, we quantify and compare the costs of process execution and adaptation of these approaches.

We employ a cost model used in our previous work [19], which incorporates two cost factors of proactive process adaptation as shown in Fig. 2. The first cost factor is *adaptation costs*, as an adaptation of a running processes typically requires effort and resources, and thus incurs costs. The second cost factor is *penalties*, which may be faced in two situations. First, a proactive adaptation may be missed. This can be due to a false negative prediction (i.e., a non-violation is predicted despite an actual violation), or because the reliability threshold was not reached, even though it was an actual violation. Second, a proactive adaptation may not be effective, i.e., the violation may persist after the adaptation.

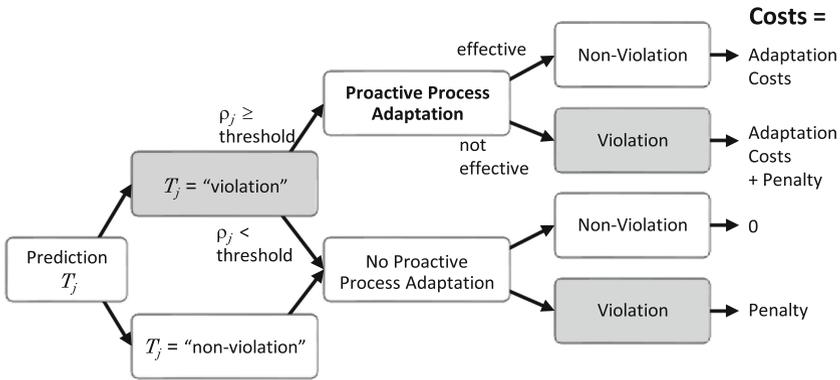


Fig. 2. Cost model for proactive process adaptation

4.2 Experimental Variables

We consider *cost* as the **dependent variable** in our experiments. For each process instance, we compute its individual costs according to the cost model defined in Sect. 4.1. The total costs are the sum of the individual costs of all process instances in our test data set. The test data set comprises 1/3 of the process instances of the overall data set.

We consider the following **independent variables** (also shown in Table 1):

Reliability threshold $\theta \in [.5, 1]$. As introduced in Sect. 3, a proactive adaptation is triggered only if the reliability of a predicted violation is equal to or greater than a pre-defined threshold, we name θ .

Relative adaptation costs $\lambda \in [0, 1]$. To be able to concisely analyze and present our experimental results, we assume constant costs and penalties (like we did in [19]). Thus, the costs of a process adaptation, c_a , are expressed as a fraction of the penalty for process violation, c_p , i.e., $c_a = \lambda \cdot c_p$. We thereby can reflect different situations that may be faced in practice concerning how costly a process adaptation in relation to a penalty may be. Choosing $\lambda > 1$ would not make sense, as this leads to higher costs than if no adaptation is performed.

Adaptation effectiveness $\alpha \in (0, 1]$. If an adaptation results in a non-violation, we consider such an adaptation effective (cf. Fig. 2). We use α to represent the fact that not all adaptations might be effective. More concretely, α represents the probability that an adaptation is effective. We do not consider $\alpha = 0$ as this means that no adaptation is effective. To reflect the fact that earlier checkpoints may be favored as they provide more options and time for proactive adaptations (see Sect. 2.2), we vary α in our experiments in such a way that α linearly decreases over the course of process execution. This means that the probability for effective proactive adaptations diminishes towards the end of the process. To model this, we define α_{\max} as the α for the first checkpoint in the process instance, and α_{\min} as the α for the last checkpoint.

Table 1. Variation of independent variables

Variable		Lower bound	Upper bound	Increment
Rel. adaptation costs	λ	0.0	1.0	.1
Reliability threshold	θ	.5	1.0	.005
Adaptation effectiveness	α_{\max}	.1	1.0	.1
	α_{\min}	.1	α_{\max}	.1

4.3 Data Sets

We use four data sets from different sources. The *Cargo2000* transport data set⁵ is the one we used in our previous work. The other three data sets are among the ones frequently used to evaluate predictive process monitoring approaches [3, 5, 32, 34, 36]. Table 2 provides key characteristics of these data sets, including the number of checkpoints we used in our experiments.

Table 2. Data sets used in experiments

Name	Pos. class	Pos. class ratio	Process instances	Process variants	Check-points
Cargo2000	Delayed air cargo delivery	27%	3,942	144	7
Traffic	Unpaid traffic fine	46%	129,615	185	4
BPIC2012	Unsuccessful credit application	52%	13,087	3,587	23
BPIC2017	Unsuccessful credit application	59%	31,413	2,087	23

4.4 Experimental Results

Using two of the data sets as an example, Fig. 3 gives a first impression of the effect of our dynamic approach. The figure shows the costs of the dynamic approach (bold) and the costs of the static approach for each of the possible

⁵ Available from <https://archive.ics.uci.edu/ml/datasets>.

checkpoints (dashed). The right hand side of each chart shows the costs without any adaptation (expressed by $\theta > 1$). These costs also serves as baseline. We chose $\alpha_{\max} = .9$ and $\alpha_{\min} = .5$, reflecting the fact that early on in the process there is a high chance that adaptation is effective, whilst at the very end, this chance is only 50%. Also, we show the results for two values of λ (relative adaptation costs). A $\lambda = .1$ reflects the situation where adaptation is rather cheap, whereas $\lambda = .4$ a situation where it is more expensive.

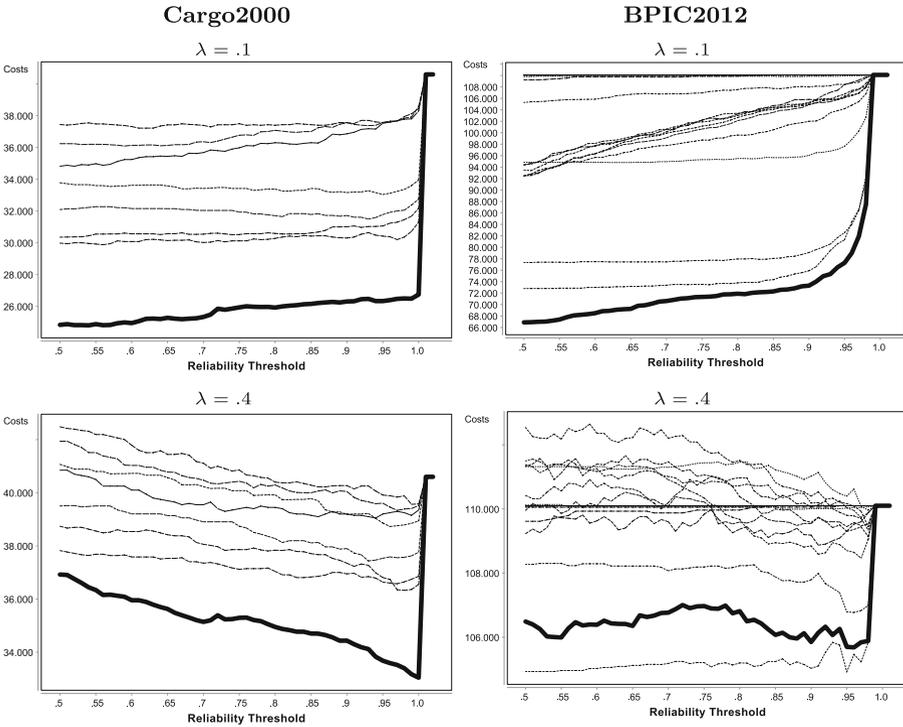


Fig. 3. Costs of dynamic approach (**bold**) vs. static approach (each dashed line represents one checkpoint); $\alpha_{\max} = .9$; $\alpha_{\min} = .5$

The charts in Fig. 3 indicate that the dynamic approach provides cost benefits when compared to the static approach, in particular when adaptations are not too expensive ($\lambda = .1$). The charts for more expensive adaptations ($\lambda = .4$) show that cost savings may become less, and may even be negative, such as for BPIC2012, where the static approach performs better for three checkpoints.

Also – independent of the static or dynamic approach – it can be observed that if adaptation costs get higher, a higher threshold (and thus more conservative stance in taking adaptation decisions) offers cost savings. While costs for $\lambda = .1$ are lowest for small thresholds, the situation is exactly opposite for $\lambda = .4$. The reason is that if adaptation costs are low (smaller λ) carrying out

unnecessary adaptations is not so costly and thus it pays off not being too conservative (i.e., setting a lower threshold). However, if adaptation costs are high (greater λ), unnecessary adaptations can quickly become very costly, and thus being more conservative (i.e., setting a higher threshold) pays off.

To further explore the situations in which the dynamic approach offers cost savings, we performed a full-factorial experiment, combining all parameter settings of our independent variables as shown in Table 1. The results for all four data sets are presented in Table 3 for different, selected values of λ . For each λ , 6666 different situations were explored.

Table 3. Number of situations (and fraction of all situations in %)

λ	A Cost(proactive) < Cost(no adapt.)		B Cost(dynamic) < Cost(static)		A - B	A Cost(proactive) < Cost(no adapt.)		B Cost(dynamic) < Cost(static)		A - B
	Cargo2000					BPIC2012				
.05	5720	85.8%	5496	82.4%	3.4%	5043	75.7%	4892	73.4%	2.3%
.25	4303	64.6%	4053	60.8%	3.8%	3443	51.7%	3189	47.8%	3.8%
.45	2353	35.3%	2030	30.5%	4.8%	1529	22.9%	1210	18.2%	4.8%
.65	704	10.6%	436	6.5%	4.0%	398	6.0%	137	2.1%	3.9%
.85	86	1.3%	14	.2%	1.1%	96	1.4%	0	0.0%	1.4%
	Traffic					BPIC2017				
.05	5696	85.4%	5282	79.2%	6.2%	5996	89.9%	5948	89.2%	.7%
.25	4385	65.8%	3971	59.6%	6.2%	5078	76.2%	5027	75.4%	.8%
.45	2673	40.1%	2280	34.2%	5.9%	3618	54.3%	3552	53.3%	1.0%
.65	560	8.4%	168	2.5%	5.9%	1739	26.1%	1661	24.9%	1.2%
.85	51	.8%	0	0.0%	.8%	31	.5%	0	0.0%	.5%

Column A shows the number of situations for which proactive adaptation leads to lower costs than the baseline costs when not performing an adaptation. This is an important metric to contextualize our results, as proactive adaptation may not be beneficial in all situations. In particular, when adaptation costs are high and the chances for a successful adaptation are low, proactive adaptation may not help (e.g., see [19]). As can be seen in column A, the relative number of situations, where proactive adaptation helps saving costs diminishes as λ increases. As an example, while for a $\lambda = .05$, proactive adaptation is beneficial in 85.4% of situations for Traffic and 75.7% for BPIC2012, this goes down to .8% and 1.4% respectively for $\lambda = .85$.

Column B shows the number of situations where the dynamic approach has lower costs than the static approach. This indicates that the dynamic approach indeed offers additional cost savings in many situations when compared to the static approach. As can be seen from the last column, the situations in which the static approach has less costs than the dynamic approach, are not very high (at around 3% on average). Again, the dynamic approach offers the highest number

of savings for smaller values of λ , where the number of situations reach 82.4% for Cargo2000 and even 89.2% for BPIC2017 to give an example.

The actual cost savings of the dynamic approach compared with the static one are shown in Table 4 for different values of λ and θ . Gray cells highlight where costs of the dynamic approach are less than the costs of the static one ('0' indicates that the costs of proactive adaptation are higher than the costs of not performing an adaptation). The table shows the cost savings for selected values of θ , averaged over all combinations of α .

Table 4. Average savings of dynamic vs. static approach in %

λ	Threshold θ										Threshold θ											
	.5	.55	.6	.65	.7	.75	.8	.85	.9	.95	1.0	5	.55	.6	.65	.7	.75	.8	.85	.9	.95	1.0
	Cargo2000										BPIC2012											
.05	24	24	23	21	21	19	19	18	17	17	16	52	52	50	49	46	45	44	43	41	34	0
.25	14	14	14	13	13	12	12	11	11	11	11	24	24	23	23	22	22	22	22	21	19	0
.45	8.1	8.4	8.6	8.4	8.4	7.7	7.6	7.4	7.3	7.2	7.3	11	12	12	12	12	12	12	12	12	11	0
.65	-14	6	6.1	6.7	6.5	5.7	4.6	4.3	4.4	4.2	4.3	2.3	3.4	3.6	4.1	4	4.2	4.8	3.6	4.1	4.6	0
.85	-20	-18	-16	-14	-12	-12	-11	-9.4	-8.3	-5.8	2.1	-28	-28	-28	-28	-27	-26	-25	-23	-13	0	0
	Traffic										BPIC2017											
.05	98	94	90	85	80	72	68	62	55	46	30	110	110	100	100	91	87	82	79	77	74	60
.25	33	33	32	32	31	30	30	29	27	25	19	44	44	43	42	40	39	38	37	36	35	30
.45	22	22	22	21	21	20	20	19	19	17	13	33	33	32	32	31	30	28	27	26	24	18
.65	-19	-18	-18	-16	-16	4.9	4.9	5.2	7.2	6	7	30	29	26	26	26	24	23	21	20	17	8.4
.85	-29	-28	-27	-26	-25	-24	-23	-22	-20	-15	-7.6	-14	-14	-14	-14	-14	-14	-13	-12	-11	-9.8	-7.2

Higher thresholds imply that cost savings are achieved for higher values of λ . As an example, for Traffic a threshold of $\theta = .6$ allows cost savings up to $\lambda = .45$, while a $\theta = .8$ allows cost savings up to $\lambda = .65$. The reason is that a higher threshold means that adaptation decisions are take more conservatively. They are taken only if a prediction is highly reliable, which in turn implies that the number of unnecessary (and costly) adaptations are reduced.

However, being conservative comes at a risk. The cost savings for higher thresholds can become smaller than the cost savings for lower thresholds. As an example, while for Cargo2000 a threshold of $\theta = .5$ leads to cost savings of up to 24%, this goes down to savings of only up to 16% for $\theta = 1$. And it may even mean that the cost of proactive adaptation is higher than not performing any adaptation, as can be seen for $\theta = 1$ for BPIC2012.

Overall (i.e, considering all possible situations), the average savings are 9.2% for Cargo2000, 27.2% for Traffic, 15.1% for BPIC2012, and 35.8% for BPIC2017. Across all four data sets, average savings are 27%. We conclude that the dynamic approach can deliver cost savings compared to the static approach, with a high chance that it is better than not performing any proactive adaptation at all.

In addition, the dynamic approach comes with the benefit that there is no need for an up-front decision on which checkpoint to use as basis for proactive adaptation, which is required in the static approach. In particular this means, that there is no need for a testing phase during which aggregate accuracies are computed in order to select a suitable static checkpoint.

4.5 Threats to Validity

Internal Validity. To minimize the risk of bias, we explored different ensembles sizes (ranging from 2 to 100). Literature indicates that smaller ensembles might perform better than larger ensembles (“many could be better than all” [38]). In our experiments, however, the size of the ensemble did not lead to different principal findings. Yet, by using a larger ensemble, we gain more fine-grained reliability estimates than by using a smaller ensemble.

External Validity. To cover different situations that may be faced in practice, we specifically chose different reliability thresholds, different probabilities of effective process adaptations, as well as different slopes for how these probabilities diminish towards the end of process execution. In addition, we used four large, real-world data sets from different application domains, which differ in key characteristics. For the sake of generalizability, we used a naïve approach to select data from the event log, i.e., we used whatever data is available and did not perform any manual feature engineering or selection. For non-numeric data attributes a categorical encoding (one-hot) was used.

Construct Validity. We took great care to ensure we measure the right things. In particular, we used a cost model that was tested in our previous work. However, we have only used constant cost functions for adaptation costs and penalties. Yet, as we showed in our previous work [18], the shape of the cost functions can have an impact on savings. We aim to investigate the impact of such non-constant cost functions as part of our future work.

Conclusion Validity. As our experiments indicate, the choice of cost model parameters (such as λ and α) impacts on whether proactive adaptation has a positive impact on cost. To address this threat, we have carefully identified influencing variables and varied them over the whole range of permissible values.

5 Conclusions and Perspectives

Dynamically determining which prediction along the process execution to use for proactive adaptation can offer cost savings. Our experimental results indicate average cost savings of 27% when compared to using a static prediction point. Such a dynamic approach thereby effectively addresses the tradeoff between prediction accuracy and prediction earliness. Also, the dynamic approach does not require a testing phase during which aggregate accuracies are computed in order to select a suitable static prediction point.

As part of our future work, we will extend our dynamic approach towards non-constant cost models. In particular, we will consider different shapes of penalties and different costs of adaptations. To this end, we will employ regression models to predict continuous indicators in order to quantify, for instance,

the extent of deviations. Regression models will also facilitate computing more complex reliability estimates (e.g., ones that use the variance of the ensemble).

Acknowledgments. We cordially thank the anonymous reviewers for their constructive comments and Richard Späker for sharing his insights of the BPIC2012 data set. Research leading to these results received funding from the EU's Horizon 2020 R&I programme under grant 731932 (TransformingTransport).

References

1. Aschoff, R., Zisman, A.: QoS-driven proactive adaptation of service composition. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSSOC 2011. LNCS, vol. 7084, pp. 421–435. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25535-9_28
2. Bosnic, Z., Kononenko, I.: Comparison of approaches for estimating reliability of individual regression predictions. *Data Knowl. Eng.* **67**(3), 504–516 (2008)
3. Breuker, D., Matzner, M., Delfmann, P., Becker, J.: Comprehensible predictive models for business processes. *MIS Q.* **40**(4), 1009–1034 (2016)
4. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
5. Evermann, J., Rehse, J., Fettke, P.: Predicting process behaviour using deeplearning. *Decis. Support Syst.* **100**, 129–140 (2017)
6. Di Francescomarino, C., Dumas, M., Maggi, F.M., Teinmaa, I.: Clustering-based predictive process monitoring. *IEEE Trans. Serv. Comput.* (2018, early access)
7. Di Francescomarino, C., Dumas, M., Federici, M., Ghidini, C., Maggi, F.M., Rizzi, W.: Predictive business process monitoring framework with hyperparameter optimization. In: Nurcan, S., Soffer, P., Bajec, M., Eder, J. (eds.) CAiSE 2016. LNCS, vol. 9694, pp. 361–376. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39696-5_22
8. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: which one suits me best? In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 462–479. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_27
9. Di Francescomarino, C., Ghidini, C., Maggi, F.M., Petrucci, G., Yeshchenko, A.: An eye into the future: leveraging a-priori knowledge in predictive business process monitoring. In: Carmona, J., Engels, G., Kumar, A. (eds.) BPM 2017. LNCS, vol. 10445, pp. 252–268. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-65000-5_15
10. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press, Cambridge (2016)
11. Gutiérrez, A.M., Cassales Marquezan, C., Resinas, M., Metzger, A., Ruiz-Cortés, A., Pohl, K.: Extending WS-agreement to support automated conformity check on transport and logistics service agreements. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds.) ICSSOC 2013. LNCS, vol. 8274, pp. 567–574. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-45005-1_47
12. Kang, B., Kim, D., Kang, S.: Real-time business process monitoring method for prediction of abnormal termination using KNNI-based LOF prediction. *Expert Syst. Appl.* **39**(5), 6061–6068 (2012)

13. Leitner, P., Ferner, J., Hummer, W., Dustdar, S.: Data-driven and automated prediction of service level agreement violations in service compositions. *Distrib. Parallel Databases* **31**(3), 447–470 (2013)
14. Leontjeva, A., Conforti, R., Di Francescomarino, C., Dumas, M., Maggi, F.M.: Complex symbolic sequence encodings for predictive monitoring of business processes. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) *BPM 2015*. LNCS, vol. 9253, pp. 297–313. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_21
15. Maggi, F.M., Di Francescomarino, C., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. In: Jarke, M., et al. (eds.) *CAiSE 2014*. LNCS, vol. 8484, pp. 457–472. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_31
16. Marquez-Chamorro, A.E., Resinas, M., Ruiz-Cortes, A.: Predictive monitoring of business processes: a survey. *IEEE Tran. Serv. Comput.* **11**(6), 962–977 (2017)
17. Mehdiyev, N., Evermann, J., Fettke, P.: A multi-stage deep learning approach for business process event prediction. In: *Conference on Business Informatics (CBI 2017)*, Thessaloniki, Greece, 24–27 July 2017 (2017)
18. Metzger, A., Bohn, P.: Risk-based proactive process adaptation. In: Maximilien, M., Vallecillo, A., Wang, J., Oriol, M. (eds.) *ICSOC 2017*. LNCS, vol. 10601, pp. 351–366. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69035-3_25
19. Metzger, A., Föcker, F.: Predictive business process monitoring considering reliability estimates. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2017*. LNCS, vol. 10253, pp. 445–460. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_28
20. Metzger, A., Neubauer, A.: Considering non-sequential control flows for process prediction with recurrent neural networks. In: *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA 2018)*, Prague, Czech Republic, 29–31 August 2018. IEEE Computer Society (2018)
21. Metzger, A., Sammodi, O., Pohl, K.: Accurate proactive adaptation of service-oriented systems. In: Cámara, J., de Lemos, R., Ghezzi, C., Lopes, A. (eds.) *Assurances for Self-Adaptive Systems*. LNCS, vol. 7740, pp. 240–265. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36249-1_9
22. Metzger, A., et al.: Comparing and combining predictive business process monitoring techniques. *IEEE Trans. Syst. Man Cybern. Syst.* **45**(2), 276–290 (2015)
23. Moreno, G.A., Cámara, J., Garlan, D., Schmerl, B.R.: Flexible and efficient decision-making for proactive latency-aware self-adaptation. *ACM Trans. Auton. Adapt. Syst.* **13**(1), 3:1–3:36 (2018)
24. Mori, U., Mendiburu, A., Dasgupta, S., Lozano, J.A.: Early classification of time series by simultaneously optimizing the accuracy and earliness. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(10), 4569–4578 (2018)
25. Mori, U., Mendiburu, A., Keogh, E., Lozano, J.A.: Reliable early classification of time series based on discriminating the classes over time. *Data Min. Knowl. Discov.* **31**(1), 233–263 (2017)
26. Navarin, N., Vincenzi, B., Polato, M., Sperduti, A.: LSTM networks for data-aware remaining time prediction of business process instances. In: *Symposium Series on Computational Intelligence*, Honolulu, USA, 27 November–1 December 2017, pp. 1–7. IEEE (2017)
27. Nolle, T., Seeliger, A., Mühlhäuser, M.: BINet: multivariate business process anomaly detection using deep learning. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) *BPM 2018*. LNCS, vol. 11080, pp. 271–287. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_16

28. Nunes, V.T., Santoro, F.M., Werner, C.M.L., Ralha, C.G.: Real-time process adaptation: a context-aware replanning approach. *IEEE Trans. Syst. Man Cybern. Syst.* **48**(1), 99–118 (2018)
29. Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y., Keogh, E.J.: Dynamic time warping averaging of time series allows faster and more accurate classification. In: Kumar, R., Toivonen, H., Pei, J., Huang, J.Z., Wu, X. (eds.) 2014 IEEE International Conference on Data Mining (ICDM 2014), Shenzhen, China, 14–17 December 2014, pp. 470–479. IEEE Computer Society (2014)
30. Polikar, R.: Ensemble based systems in decision making. *IEEE Circ. Syst. Mag.* **6**(3), 21–45 (2006)
31. Poll, R., Polyvyanyy, A., Rosemann, M., Röglinger, M., Rupprecht, L.: Process forecasting: towards proactive business process management. In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) BPM 2018. LNCS, vol. 11080, pp. 496–512. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98648-7_29
32. Rogge-Solti, A., Weske, M.: Prediction of business process durations using non-markovian stochastic petri nets. *Inf. Syst.* **54**, 1–14 (2015)
33. Salfner, F., Lenk, M., Malek, M.: A survey of online failure prediction methods. *ACM Comput. Surv.* **42**(3), 10:1–10:42 (2010)
34. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30
35. Teinemaa, I., Dumas, M., Maggi, F.M., Di Francescomarino, C.: Predictive business process monitoring with structured and unstructured data. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNCS, vol. 9850, pp. 401–417. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45348-4_23
36. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: review and benchmark. *CoRR* abs/1707.06766 (2017)
37. Weber, B., Sadiq, S.W., Reichert, M.: Beyond rigidity - dynamic process lifecycle support. *Comput. Sci. R&D* **23**(2), 47–65 (2009)
38. Zhou, Z.H.: *Ensemble Methods: Foundations and Algorithms*. Chapman and Hall/CRC, Boca Raton (2012)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

