Machine Learning Algorithms for Predictive Business Process Optimization: An Evaluation Study

Dhirendra Negi
Research scholar
Department of Computer Science
Singhania University

Dr. Ajay Agarwal
Supervisior
Department of Computer Science
Singhania University

Abstract: Predictive Business Process Monitoring (PBPM) using machine learning algorithms has emerged as a powerful strategy for enhancing decision-making in organizations. This study explores the application of PBPM in a real-world context, focusing on the potential cost savings and efficiency improvements compared to traditional human decision-making processes. Using a synthetic event log of the peer review process for scientific publications, we assess the feasibility and effectiveness of PBPM by considering various thresholds for automated acceptance or rejection. Our findings demonstrate that machine learning approaches hold promise in the realm of business process management, offering insights into the potential financial impact of predictive analytics.

Keywords: Predictive Business Process Monitoring, Machine Learning Algorithms, Decision Support, Process Optimization, Cost Savings, Efficiency Improvement

1. INTRODUCTION

Predictive Business Process Monitoring (PBPM) using machine learning algorithms is a critical strategy for improving decision-making in businesses and organizations. This approach involves making predictions about the outcomes of various business processes, which can lead to cost savings and efficiency improvements. However, the implementation of PBPM must consider the balance between early predictions and the risk of errors, as human decision-makers may have to intervene in case of inaccuracies. This study explores the PBPM framework and its application in a real-world example, the peer review process of scientific publications, to demonstrate the concept.

2. OBJECTIVES OF THE STUDY

To assess the cost-effectiveness and feasibility of implementing predictive business process monitoring (PBPM) using machine learning algorithms in a real-world business context, specifically by analyzing the potential cost savings and efficiency improvements compared to traditional human decision-making processes.

3. RESEARCH METHODOLOGY

Predictive Business Process Monitoring Framework

The deciding factor for a person in a decision-making role in any business or organization is, of course, the issue of whether or not the Predictive Business Process Monitoring strategy offers any benefits to the company or organization. In point of fact, there is no simple response to this inquiry at all. The following two requirements must be satisfied for any predictions made using an algorithm for machine learning to make any sense at all:

The forecasts cause some form of activity to take place.

They are carried out at a level that is sufficiently early in the process that human decision makers have, in most cases, been unable to anticipate the end result of the process.

If it weren't the case, the human being making the choice would do what has to be done nevertheless. To take action based on information that was generated by an unintelligible black box at a moment in time that is too early for human decision makers to foresee the end result of the process certainly carries with it a significant amount of risk.

The PBPMF applied to a simple example – The Review Process

We wish to use the peer review process of a scientific publication as a model for this context so that it may serve as an example. Figure 3.2 presents the model of the process further down the page.

The example was developed in a partnership between the Technical University of Delft, the University of Twente, and the Technical University of Eindhoven (3TU.Datacentrum). It is included with the data from the event log, and it was supplied by Wil van der Aalst from TU Eindhoven in the year 2010. Since the processes that occur in the actual world are, for the most part, far more complex, it may make sense to insert a great number of extra context-specific properties and effects into the model. This straightforward method, on the other hand, is all that is required to determine whether or not the model is applicable. The review procedure is kicked off when a work is submitted to the journal for consideration. After that, it will always be given to three different people to evaluate. There are three different replies that might be given by each of the three reviewers as follows:

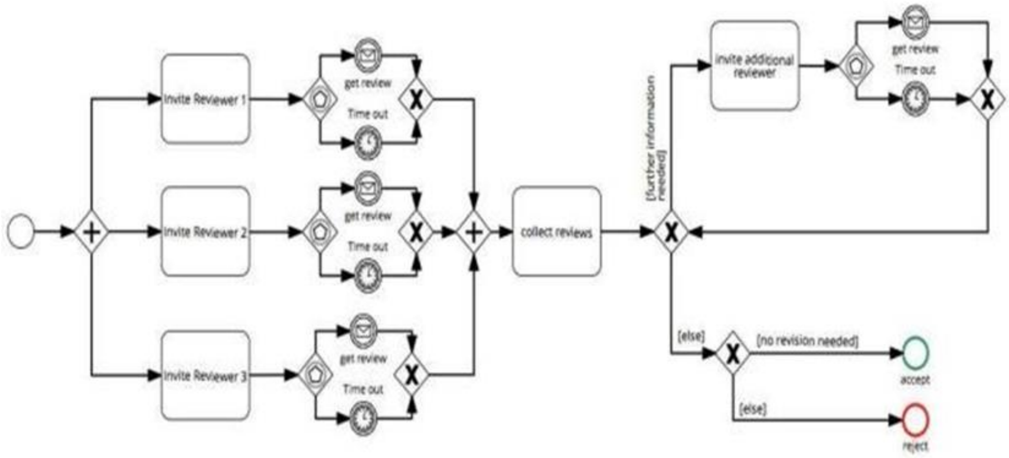


Figure 1: The Review Process.

either a request to accept the paper, a suggestion to reject the document, or a time-out, the latter of which occurs in the event that a certain deadline is exceeded.

After the editors of the journal have received comments from all three reviewers (in this context, time-out is considered to be one of the available responses), they determine whether or not they need more information. If this is the case, the manuscript will be allocated to an extra reviewer. Here again, all three replies stated above are feasible. This process-loop is continued until the editors consider that they have enough information to reach a conclusion. Then, the document is eventually either approved or denied.

The Variable Prediction Point

Our strategy is to hasten the completion of this procedure as its primary objective. As a result, an early step in the process will include the establishment of a prediction point. After the editors have obtained the responses from the three people responsible for the initial review, we have reached the earliest possible stage in our exemplary process. After the completion of the first extra reviewer loop, which comes one step later in the process, there will be another time at which probable predictions may be made. Therefore, the prediction point could be set after n additional reviewer loops are finished, where n is an integer greater than or equal to zero.

At this point in time, the algorithm is provided with all of the information that was generated during the course of the lifespan of the process that is presently being carried out (which is assumed to be already trained at this time). After that, the algorithm makes a forecast about that particular process instance, specifically, in this example, regarding whether the ultimate result will be a accept or a reject. It is possible to take action based on this prediction: If the forecast indicates accept (reject), the paper might be immediately accepted (rejected), so eliminating the possibility of iterating again through the extra review loop and, as a result, saving money.

The Cost Variables

As was mentioned earlier, the upside potential for cost savings is accompanied by a downside potential for possible errors in the prediction. The paper could be directly accepted (rejected), based on the algorithm's prediction, whereas the human decision makers would have sent the paper through several more additional review loops and finally come to the conclusion that the paper should be rejected. The downside potential for possible errors in the prediction is as follows: the upside potential for cost savings is accompanied by the downside potential for possible errors in the prediction (accepted). We define an error as any departure from the result that would have been obtained by a human decision maker had they been responsible for the choice.

The model that integrates the qualities that we specified above will be presented in the coming part of this article.

As a result, we will define the following three variables:

r: represents the cost of going through one cycle of the further review loop.

The forecast leads to the rejection of a manuscript that, in reality, should have been approved. This is the cost of a type-1 mistake.

The forecast results in the acceptance of a paper that, in reality, ought to have been rejected. This is the cost of making a type-2 mistake.

If we have access to a data collection that contains historical process occurrences, we may conduct an analysis to determine whether or not our PBPM technique is likely to result in financial savings. This estimation can be achieved by contrasting the total costs that would have been incurred for the specified set of instances if they had been run by human decision makers with the respective costs that are incurred when decisions are (partially) taken based on the algorithm's prediction. This will give an accurate representation of the amount of money that could have been saved by using the algorithm.

It is possible to simplify this situation by setting the overall costs for the human-run processes to zero. This will ensure that sufficient attention is paid to any cost variation, including cost savings from the elimination of review loops and extra costs incurred as a result of poor judgments. In order to provide a concise illustration of the method outlined above, we will apply it to a modest collection of historical data that is composed of the four process cases shown in Table 1. In the second column, the frequency with which the manuscript was subjected to the further review is detailed for each case. In the third column is shown the final choice that was finally made by human decision makers about whether or not to accept the paper. The choice that would have been made based on the algorithm's forecast is shown in the fourth column of the table.

We are able to calculate the Cost Deviation for each instance using the information contained in these three columns by first multiplying the number of times the algorithm was run through the additional review loop by r and then subtracting f1 and f2 for any instances in which the algorithm's prediction was incorrect and consequently resulted in a type-1 or type-2 error.

Table 1: Cost Deviation comparing human decision with algorithm's decision.

ID	#additional review loops	Human decision	Algorithm's decision	Cost deviation
1	2	Accept	Accept	$2r$
2	5	Reject	Reject	$5r$
3	7	Accept	Reject	$7r - f_1$
4	3	Reject	Accept	$3r - f_2$

Introducing Thresholds for Automated Acceptance or Rejection

Machine learning classifiers are able to conduct the classification of an item, as well as give a class probability to each object, which expresses the degree of conviction about the classification. In addition, machine learning classifiers are able to do the classification of an object. This gives us the ability to specify two thresholds, which we will refer to as Taccept and Treject. If the degree of assurance that the algorithm has for accepting (or rejecting) the article is greater than Taccept and Treject, the paper that is now being examined is then directly accepted (or rejected). If the degree of confidence generated by the algorithm is lower below the thresholds, the prediction will not result in any action being taken.

In the event where the total of both thresholds is less than one, it is possible that both thresholds be surpassed at the same time. In this instance, decisions are being made based on the forecast with a better degree of confidence in order to take appropriate action. The sort of action that is being performed is shown in Table 1, and it varies based on whether or not each threshold has been achieved.

Table 2: Action taken based on reached thresholds

T_{accept} reached	T_{reject} reached	Action
TRUE	FALSE	Directly accept paper
FALSE	TRUE	Directly reject paper
FALSE	FALSE	Let human decision-maker continue to run process
TRUE	TRUE	Take action based on prediction with higher certainty

The issue now that has to be answered is what values need to be picked for the two thresholds. If the criteria are set very high, such as 0.99, we may anticipate seeing fewer mistakes, which will allow us to avoid huge expenses associated with making poor selections. On the other hand, it is likely that a degree of confidence greater than 0.99 will only be achieved very seldom, which will result in only marginal cost reductions owing to the elimination of review loops. On the other end of the spectrum, the thresholds may be set as low as 0.51 for example (Note, that the certainties for acceptance and rejection add up to one, thus a value below 0.5 for the rejection certainty implies a level of certainty above 0.5 for acceptance and vice versa.). This would result in significant cost savings due to the elimination of a large number of review iterations, but on the other hand, significant expenses would be paid due to the large number of incorrect choices that would be made.

Translation into an Optimization Problem

The difficulty of determining how the thresholds should be set translates into an optimization problem with two dimensions: Select the appropriate values for Treject and Taccept in such a way that the Cost Deviation, which is based on a specific collection of historical data, is maximized:

$$(T^{*}_{reject}, T^{*}_{accept}) \coloneqq argmax_{[0,1]x[0,1]}\{Cost\ Deviation_{r,f_1,f_2}(T_{reject}, T_{accept})\}$$

The third variable that may be altered is where to position the forecast point, which can also be thought of as an independent variable. The inclusion of the prediction point as a new variable leads to a three-dimensional optimization of the Cartesian product [0, 1] x [0, 1] x N0, which results in a maximum value.

It is vital to keep in mind that the choice of where to place the forecast point is a significant one for the following three reasons:

To begin, it should be understood that the algorithm can only make use of data that was produced during the current process instance up to the point when the prediction is being made. When this point is set later, more information will have been produced throughout the lifetime of the relevant process instance. As a result, the algorithm will be able to assess more data in order to do its prediction, which should typically result in an improvement in accuracy. Because of this concern, a late prediction point is promoted.

On the other side, late prediction scores come with two potential drawbacks:

A forecast that is conducted at a late point of the process is often less valuable and has less potential for cost savings than one that is performed earlier in the process.

A late forecast point may also indicate that there is less historical data available to train the Machine Learning system on in some circumstances.

This is also the case in the scenario that we will review: For example, only 43 of the past process instances out of a total of 10,000 had completed 15 additional review loops. So, the algorithm only has only 43 examples to learn how to utilize the whole data that has been produced up to this stage in the prediction process. Instances that did not complete until after a certain number of iterations, say 14, could be added to the training data, but from those, the algorithm can only learn how to use the information that was created up to the 14th review loop, which then implies that the prediction could be performed after the 14th round anyway, without any loss in accuracy. Of course, instances that did finish after a certain number of iterations could be added to the training data.

Cost Deviations for Different Threshold Combinations

We can discretize its intervals [,] through a grouping in order to find the optimum for a given prediction point and given cost variables. Then, we can iterate over all threshold combinations in these partitions, and for each combination, we can calculate the Cost Deviation based on historic data. This will allow us to find the optimum. There are a few key aspects that should be brought to your attention:

The historic data that is utilized to do this calculation must be the test data set; it cannot be the data set that was used for training. If it were, the computation would not be accurate.

The Cost Deviation that is generated in this manner only provides a forecast for the future that is based on data from the past. So, the optimization process has to be carried out on a consistent basis using data that is more current.

Reducing Computational Complexity of the Optimization

Computing time can be reduced by some logical considerations:
If both thresholds are in $[0,0.5]\times[0,0.5]$, then at least one threshold has to be reached, since the certainties sum up to one. Let's assume w.l.o.g. that T_{reject} is reached and T_{accept} is not reached. Then, the certainty for "reject" has to be above 0.5. (If not, then $Certainty(\text{"accept"})$ which is $1 - Certainty(\text{"reject"})$ has to be above 0.5, but then T_{accept} would have been reached since it is in [0,0.5].)
If both thresholds are reached, then the prediction with higher certainty is used, and this has to be above 0.5 for the same reason. So for all threshold combinations in $[0,0.5]\times[0,0.5]$ the rules can be reduced to "Take action based on the prediction with certainty above 0.5", and this rule does obviously not depend on the values for the thresholds.
Subsequently, the number of errors being made for respective threshold combinations, as well as the number of loops avoided, can be calculated by using a single threshold combination within this range, rather than iterating over all 2500 combinations (assuming a step size of 0.01 here).
Similarly, for $T_{reject} > 0.5$, the value of T_{accept} does not have any impact on the number of false rejections (type-1 errors): If the certainty for "reject" is greater T_{reject} which is greater 0.5 itself, it does not matter whether T_{accept} was reached as well: If it was not reached, we will reject. If it was reached as well, the certainty for accept has to be below 0.5 since the certainty for reject was above 0.5, so we will reject again. So, for the calculation of type-1 errors, we don't need to iterate over different values for T_{accept} where T_{reject} is greater 0.5. The reversal is true for the calculation of type-2 errors. This reduces the number of threshold combinations from 100 times 49 to just 49.

There are further logical implications that may be derived that are comparable to the one that was presented before. It is possible that it is crucial to take advantage of these simplifications if the environment is extremely dynamic and requires the optimization to be done on a very regular basis. This is not a particularly relevant consideration for the typical use case, in which the optimization is executed maybe once or twice a year; hence, we do not go into more depth on this subject in this paper.

Implementation

We have completed the implementation of the model that was discussed before and have carried out a series of tests by making use of the synthetic event log of the article review process that was shown earlier. 10,000 different process instances and 236,360 different occurrences are included in the synthetic event log. Since this is an artificial data set and there was no information regarding the process that was used to produce the data, it was not possible to determine whether or not the data would include any patterns that would suggest a greater chance of one particular event over another.

As a result, we changed the process outcome (accept or reject) in the data set such that it reflected our adjustments. The new categorization uses a variety of different rule sets, each of which pertains to a different property of the individual process. In order to categorize the process occurrences, eight distinct sets were used. Please refer to the appendix for more information on these changes and their specifics.

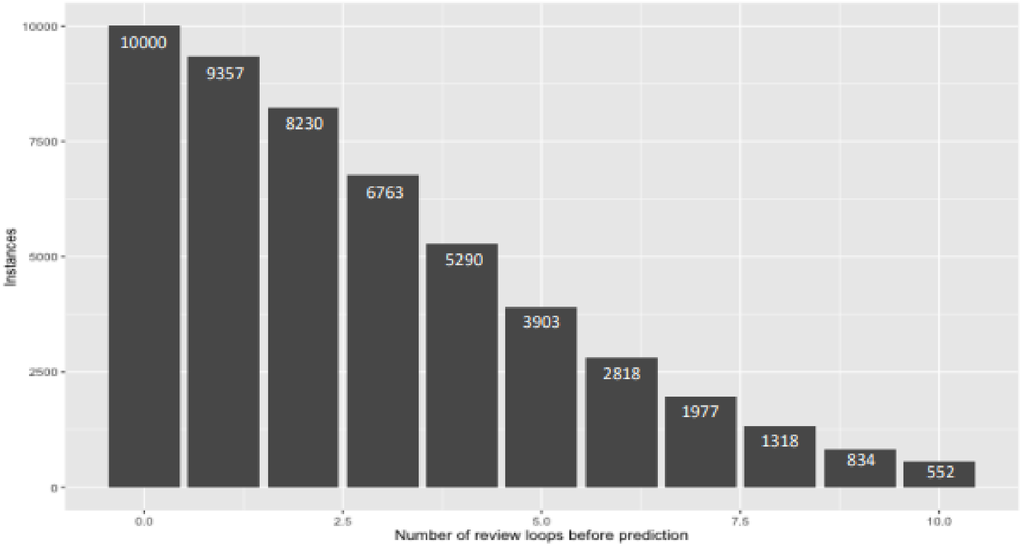
Steps in Pre-Processing

The use of machine learning that is known as predictive business process monitoring is quite specialized. The procedures that we had to take in order to be able to use supervised machine learning methods for our prediction job are outlined in the next portion of this article. Given that the dependent variable may have the attribute of either accept or reject, the objective of this prediction job is to accurately identify process occurrences. In order to recreate the model of the business process, the first thing that we did was look at the many variables that were included inside our data.

The next step required us to choose which aspects of the predictions We would think about it carefully. Due to the simplicity of the structure, we can enumerate all possible prediction points by writing,..., where I is the index of the additional review loop that led to the establishment of the prediction point (the maximal number of review loops was 25 in our data set). Nevertheless, adding late prediction points in the experiment scenario is not warranted since the number of cases that reach a given review loop rapidly decreases: Just 43 of the 10,000 process instances in our data set have progressed through 15 more review loops, which is insufficient for training a machine learning system.

Figure 2: Training/testing sample size in terms of prediction points.

Figure 2 depicts, for each review loop, the number of instances that have reached the end of the process and are now eligible to be used in training for a prediction point. Given the little information available for such forecast points, we made the heuristic decision to ignore them going forward. After more than a decade of deliberation, we've come to this conclusion.



The most time-consuming part of our method was the initial data preparation stage. Our first step was to reformat the synthetic log file such that each row would represent a single running instance of a process, each with its own own process ID. We modeled dummy variables for each step of the process and got rid of the ones that didn't add anything to the prediction work. We also calculated the time differences between two points in time to generate extra variables that account for the duration of certain process stages. At the end of the day, for each forecast point, the data set must be altered such that it only includes records of events that happened before to that point. In data mining parlance, false predictors or leaks refer to variables that correspond to events that occurred after the forecast point and must be eliminated for each individual prediction.

4. ANALYSIS AND RESULT

Accuracy of Different ML Techniques as a Function of the Prediction Point

We decided to work with seven distinct classification methods so that we could provide an overview of the applicability of a variety of machine learning (ML) classifiers. This was done using various sets of classification algorithms. Since there are so many different algorithms to choose from, deciding which one to use may be a challenging and time-consuming process, as was said previously. We did not compare the performance of each and every classifier because our primary focus was not to maximize classification performance on this particular data set to the very last bit, but rather to introduce a generalizable framework. Because of this, rather than comparing the performance of each and every classifier, we tried to include all relevant and state-of-the-art algorithms in our selection. Classifiers such as Decision Tree, Random Forest, AdaBoost, Bagging, Neural Network, Logistic Regression, and Support Vector Machine were among those with which we worked. We have decided not to implement NN because of the substantial assumptions that this algorithm makes on the independence of characteristics; these assumptions led us to this decision. The seven different classifiers each had their parameters tuned using methods that were considered cutting edge at the time.

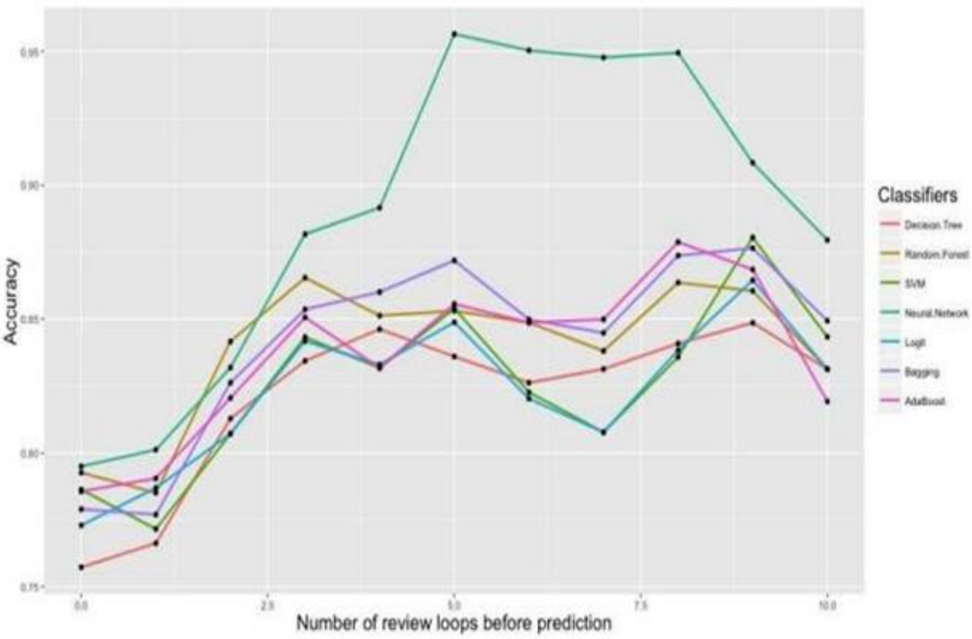


Figure 3: Accuracy of the employed ML algorithms per prediction point.

Next, we trained the classifiers and checked the accuracy of their predictions using a test set. To do this, we divided the data into three parts, with two thirds serving as training data and the other third being held back for testing. The accuracy score was selected as the assessment technique because we are particularly interested in the amount and kinds of mistakes that the classifier creates in comparison to the total number of instances that it has categorised. This is why we opted to evaluate it using this approach. Figure 1 depicts the outcomes of the experiment.

To begin, there is a discernible trend toward ever-increasing precision across all classifiers in time steps leading up to a maximum of three review loops before prediction. This demonstrates the predicted association between the quantity of data that is available for training and the accuracy achieved on cases that have not been seen before. Nevertheless, another thing that we discovered was that the mean accuracy score remained the same for subsequent prediction points and even slightly reduced as the study came to a close. Figure 2 depicts the mean accuracy score across all classifiers for each prediction point, together with a confidence interval shown as a shaded area for 95%.

We hypothesized that this impact was caused by the decreasing number of cases that were available for training at later prediction points, and we verified this assumption by generating additional instances for each of the prediction points in question and then rerunning the classification process. The accuracy ratings that were obtained as a consequence were consistent with the hypopaper that accuracy would improve as more data was collected (which equals later prediction points). As a result, we are able to draw the conclusion that the impact happens as a result of a lack of data.

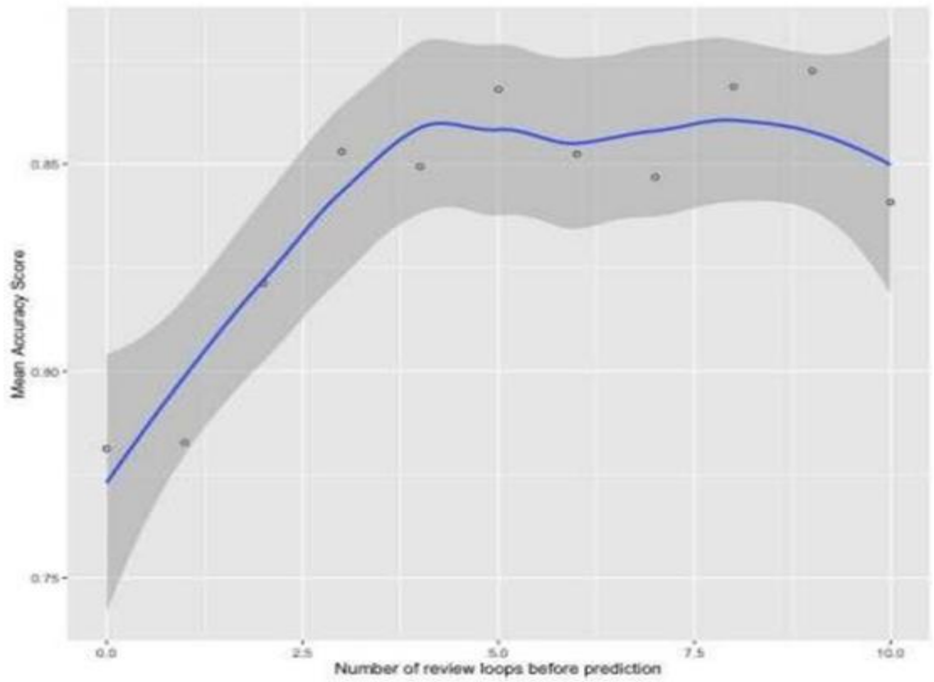


Figure 4: Mean accuracy over all classifiers per prediction point.

In reiterating these results, we come to the conclusion that there is a considerable trade-off that has to be taken into account and needs to be addressed before the PBPM method can be used as a tool for supporting decision making. When there is a clear trade-off between lower costs incurred as a result of an early forecast and greater accuracy scores incurred at later prediction points (which also result in lower costs incurred as a consequence of error reduction), one has to locate the prediction point that is ideal. The three-dimensional optimization problem that was discussed before is being used in order to find a solution to this issue.

When it came to practically all of the different prediction points, the neural network that we used had the best performance of any other classifier. This neural network model would most likely be selected as the default classifier in the case of a traditional machine learning issue, where the primary objective is to optimize performance on examples that have not been seen before. Nevertheless, since we are dealing with a supervised classification issue in an organizational setting, there are a few extra characteristics that need to be taken into consideration in order to be able to discover the model that is best appropriate for the undertaking.

Therefore, the accuracy score continues to be one of the most essential criteria, since any decision-support technology that results in fewer incorrect judgments is the one that should be favored. But in addition to that, it is desirable for the model that was produced to have a high level of interpretability. Businesses undoubtedly desire to have the ability to comprehend why a decision support tool behaved in a certain manner and what the primary variables of influence were for the conclusion that it made. Another factor that plays a role in determining the total value of a machine learning classifier for an organization is the amount of computational resources it requires.

These criteria have to be selected on an individual basis, and they are, of course, the product of needs that are quite job specific. Nevertheless, we would want to underscore the fact that ML that is applied to a business context has to meet several criteria, and it is not always feasible to determine the optimum algorithm for a job by looking at conventional performance indicators. This is something that we would like to emphasize.

Number of Errors as a Function of the Thresholds

For the sake of our testing, we made the decision to concentrate entirely on the accuracy and to disregard the issue of interpretability. As a result, we went with the method that had the best degree of precision for each forecast point (which was always the neural network, except for a prediction point after two add. review loops). After applying the chosen machine learning algorithm to the training data set, the result was a prediction for the categorization of each occurrence, along with the degree of confidence associated with each prediction. Because of this, we were able to cycle through all possible threshold combinations in the Cartesian product PP, where P stands for the partition.

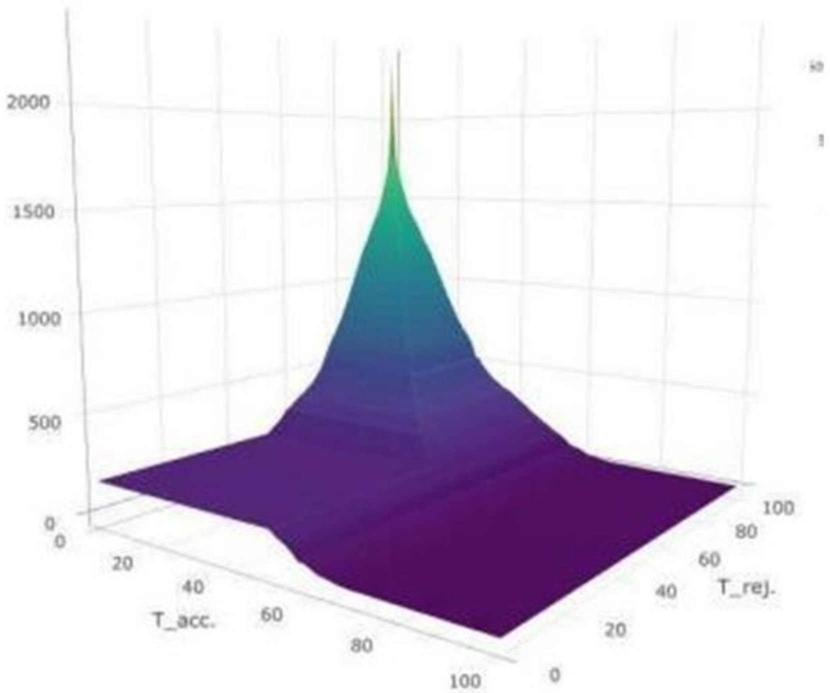


Figure 5: Type-1 Errors.

A surface plot depicting the number of type-1 mistakes for each possible combination of reject and accept thresholds for PP may be seen in Figure 4.4. Once there were no more loops of reviews done, the prediction point was decided upon. There will be a peak of 2359 type-1 mistakes if the combination of Treject and Taccept is chosen such that it is very near to zero and very close to one, respectively. This will cause practically all papers to be rejected directly. Since the total number of certainties always adds up to one, any combination in the range [0, 0.5][0, 0.5] will always cross at least one threshold. For these threshold combinations, the action to take is determined by the forecast that has a greater degree of confidence. As was to be predicted, the highest Treject threshold was associated with the lowest count of type-1 mistakes observed. As was previously said, the level of Taccept does not have any impact on Treject values greater than 0.5.

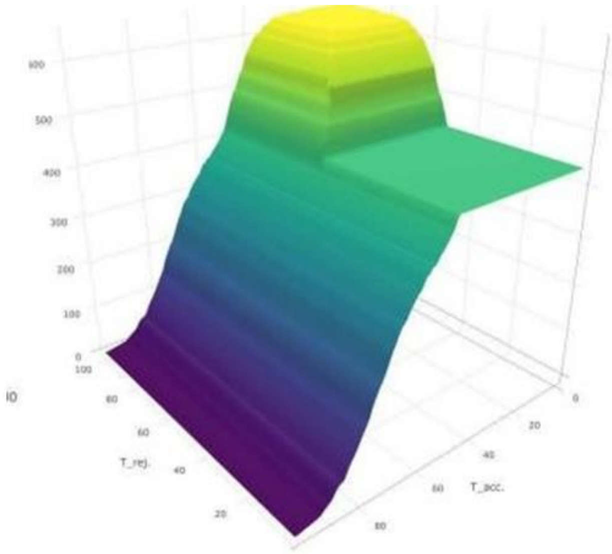


Figure 6: Type-2 Errors.

Figure 5 displays the total number of type-2 mistakes that were found. To no one's surprise, the combination of a high rejection threshold and a low acceptance threshold is where it reaches its maximum value. (It is important to take into account that the point of view has shifted for the purpose of this graph.) In this worst-case scenario, every single article is approved, which results in 641 erroneous acceptances. The total number of occurrences in our test data set is 3000, thus adding the maximum of 2359 type-1 errors brings that total up to the desired value.

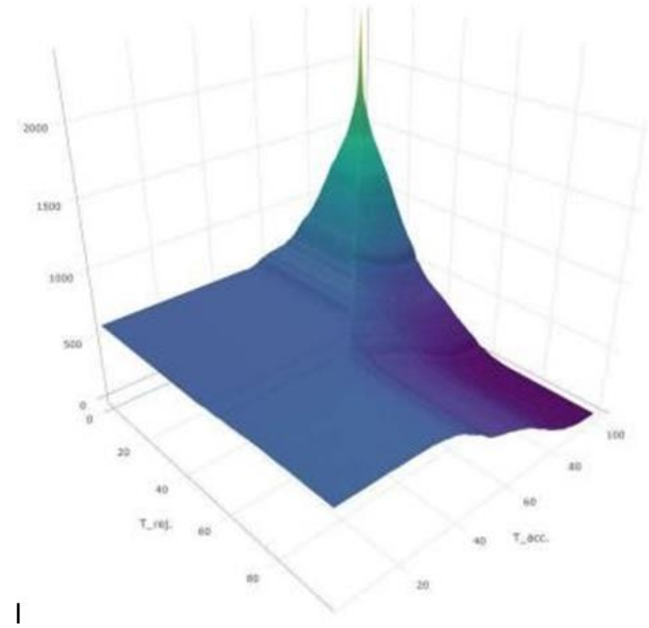


Figure 7: Sum of type-1 and type-2 errors.

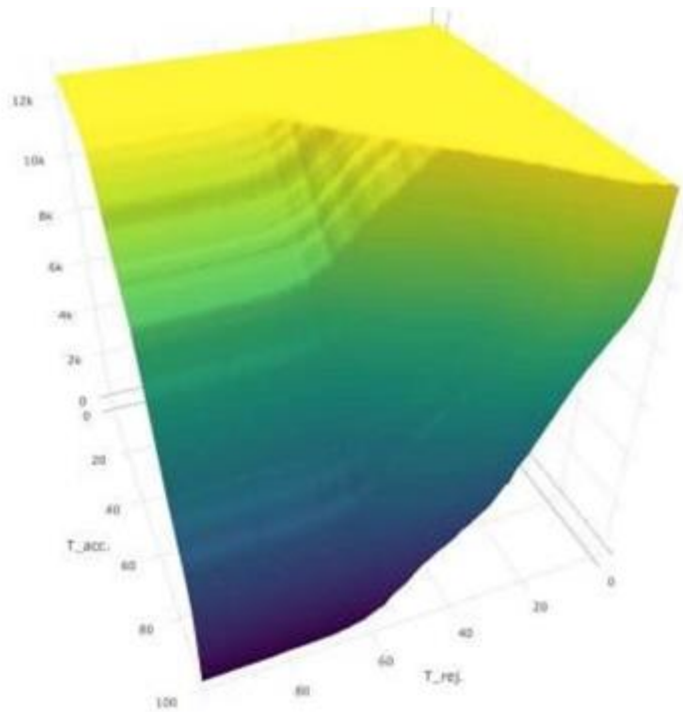


Figure 8: Number of loops avoided.

Number of Loops Avoided as a Function of the Thresholds

The amount of unnecessary loops is shown in Figure 7, which may be seen here. This figure might be as low as 12,000 for relatively low thresholds or as high as zero for very high levels of risk. We need to set values to the cost variables in order to get rid of the trade-off that exists between the advantageous impact of having a large number of saved loops and having a high mistake rate. We tried out a few different combinations of the cost variables in our tests. In order to take into account the greatest possible number of occurrences, the prediction point was determined to be zero loops after additional reviews. The outcomes for further prediction points seem to have a similar framework.

Economic Results – Cost Deviations in Different Scenarios

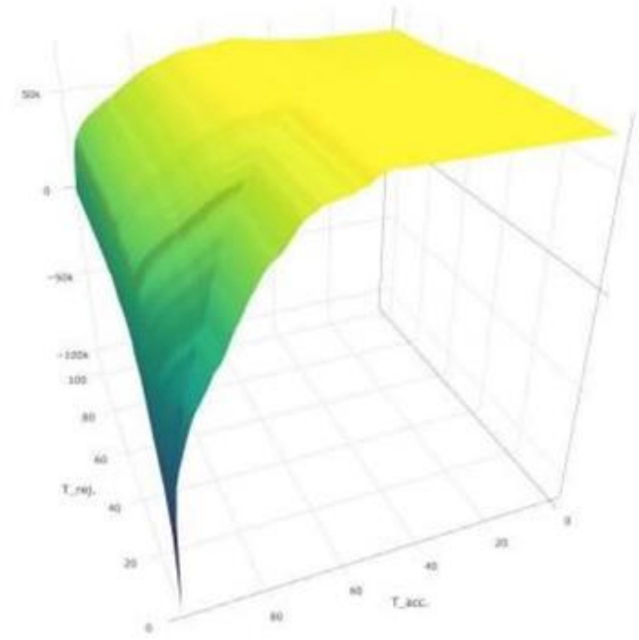


Figure 9: Cost Deviation in Scenario 1.

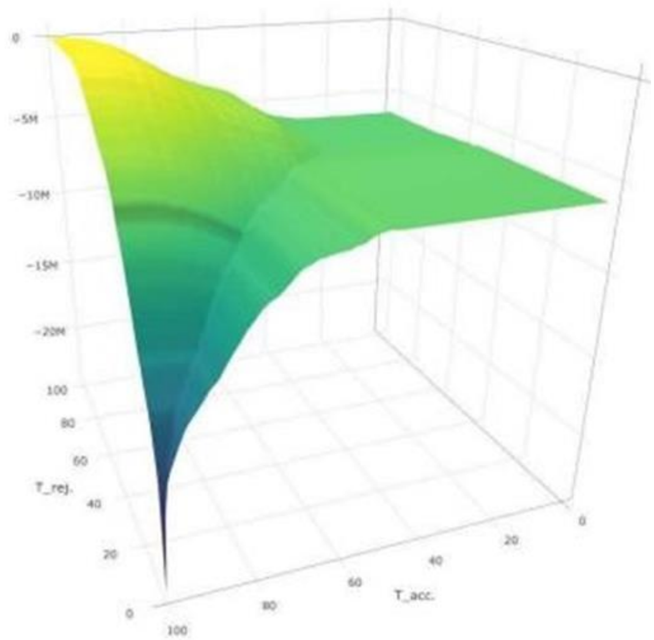


Figure 10: Cost Deviation in Scenario 2.

Scenario 2 (Figure 11)**:**

$f_1 = 10{,}000, f_2 = 10{,}000, r = 0.1$

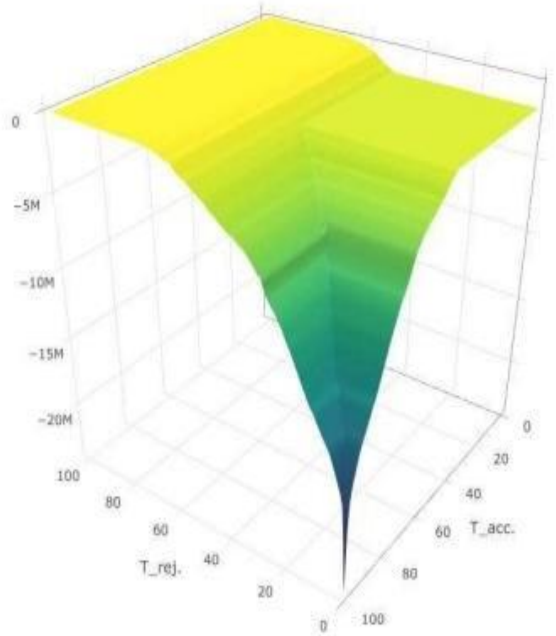
This is an extreme scenario where errors are extremely expensive relative to the costs for an additional review loop. Values like this could make sense e.g. in a medical setting, where errors could be life-critical.

The maximum here is 241 which is reached for $T_{accept} = 0.99$ and $T_{reject} > 0.98$. In such a setting, with very expensive mistakes, the application of our prediction approach seems to be questionable. On the other hand, one has to keep in mind that errors are measured against what a human decision maker did decide for a certain instance. It is not clear whether the human decision really was the best one. If possible the measurement could be performed against an objective ex-post classification.

Scenario 3 (Figure 12)**:**

$f_1 = 10{,}000, f_2 = 100, r = 10$

The maximum of 68,390 is reached for $T_{accept} = 0.52$ and $T_{reject} > 0.98$.



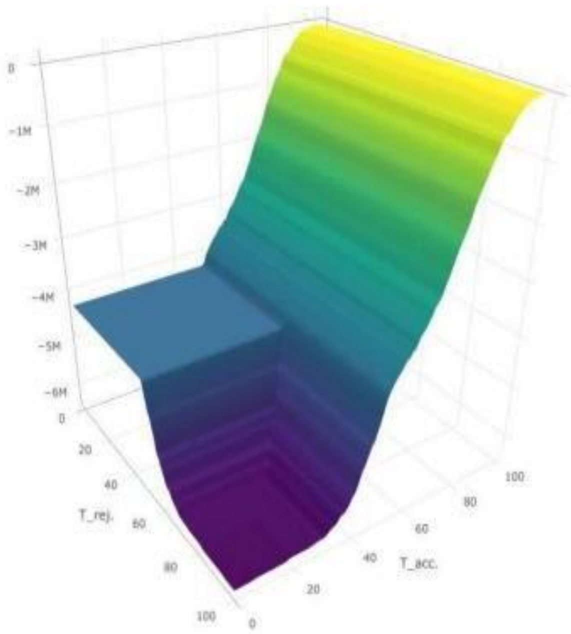


Figure 11: Cost Deviation in Scenario 3.

Figure 12: Cost Deviation in Scenario 4.

Scenario 4 (Figure 13)**:**

$f_1 = 100, f_2 = 10{,}000, r = 10$

The maximum of 24,750 is reached for $T_{accept} = 0.99$ and $T_{reject} = 0.68$.

Average Cost Deviation per Instance

In our environment, the best prediction point was always the earliest feasible (after zero loops of further reviews), and this was true regardless of the values that we gave the cost variables. But, in addition to the overall Cost Deviation, one can also be interested in looking at other performance data. For example, one would be curious about the typical Cost Deviation for each individual instance (which is not just proportional to the total Cost Deviation, since the number of instances reaching a certain prediction point varies with the prediction point). We investigated the connection between the average deviation across all instances and the cost ratio,, in which stands for the cost of each extra review loop and = = is the cost of making a mistake. The charts for all of the prediction points from 0 to 10 are shown in Figure 13. When seen from a distance, it seems that the earliest forecast point is the most accurate in this case as well; but, when viewed up close, as shown in Figure 14, this perception shifts. When the cost of mistakes is enormous in comparison to possible savings, the better accuracy for later forecast points becomes more important. This is the case when the ratio of to is very low. Because of this, the forecast point that has through five extra review loops, which is the one that was found to have the greatest accuracy, is the one that has the lowest average cost deviation for each occurrence.

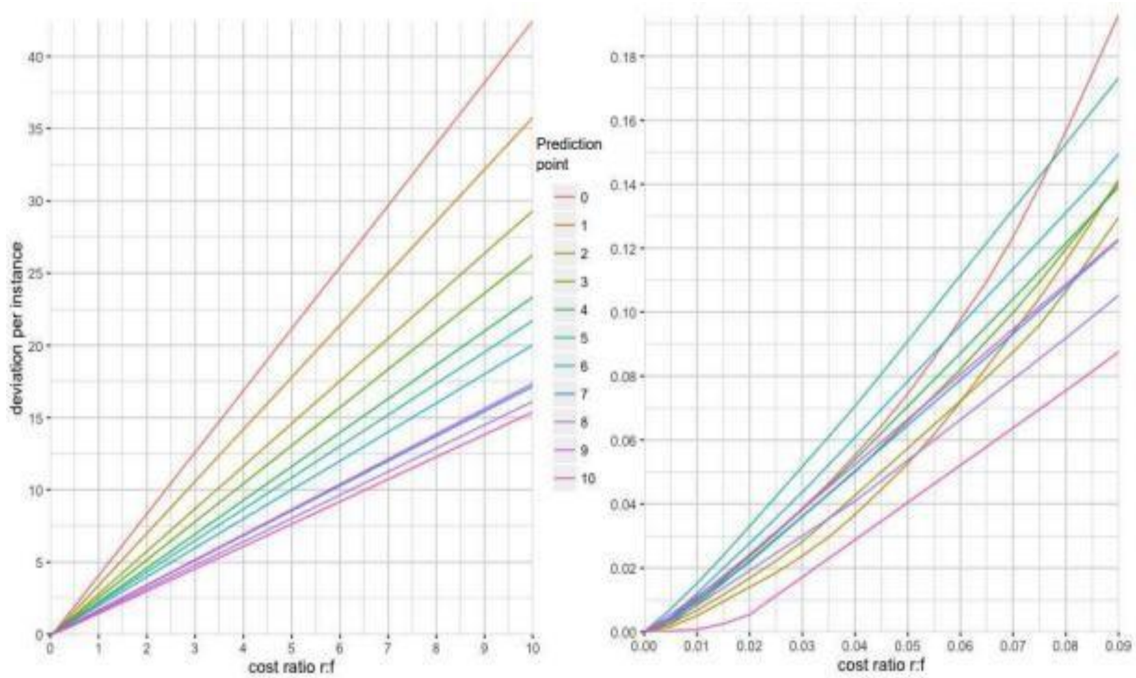


Figure 13: Avg. Cost Dev. per instance **Figure 4.14: Zoom into Figure 14.**

Recapitulation – Necessary Steps in the Evaluation and Optimization Process

In order to provide a comprehensive summary of the assessment and optimization process, the following procedures need to be carried out in an actual environment:

Determine the accuracy of various machine learning algorithms based on the prediction point, and then make a decision on which algorithm to employ and when based on the results of this measurement. (If desired, add concerns regarding interpretability into the option).

Let the algorithm of your choice to execute the classification on the test data set for each individual prediction point. Make it a point to check that the algorithm, in addition to the categorization, also includes its degree of confidence (aka class probability).

For each prediction point, iterate over all possible threshold combinations and determine the total number of type-1 mistakes, type-2 errors, and process steps that were skipped (in our case: number of additional review loops). (The simplifications which we outlined in the paragraph Reducing Computational Complexity of the Optimization may be implemented here.).

Give the cost variables their respective values. Calculate the Cost Deviation for each possible combination of thresholds for each individual forecast point. Find the best solution at each of the forecast points.

Evaluate the optimal solutions for each of the prediction points, and choose the one that yields the highest value. This results in the best possible prediction point as well as the best possible threshold combination for each individual prediction point.

Discussion and Managerial Implications

The findings that were reported are very restricted to the particular data set that was utilized for the examination. Even if neural networks have shown high performance, this does not mean that this algorithm is always the most effective or that it should be employed in the context of business operations. It is not within the purview of that study to provide a suggestion about which method of machine learning should be used in which particular setting. In addition, the effectiveness of the various algorithms is heavily reliant on characteristics of the specific data set in question as well as the fine-tuning of the relevant parameters (King et al. 1995).

As a result, making a broad claim about which algorithm is the most effective is neither realistic nor attainable. However, we do not intend to center our attention on a particular application domain or a process in particular. Our goal is to build a framework that can be used to a wide variety of processes, as well as to show that the suggested model is applicable to real-world situations. After the review, it is now clear that Machine Learning may be used in the context of BPM to anticipate the performance of the process. The method describes in detail which machine learning techniques can be used in the context of BPM with regard to the specific characteristics of process logs and the specific characteristics of the various algorithms. When deciding which classifiers to use in a certain situation, it is important to first conduct an analysis of the relevant individual traits and needs. In a similar vein, the choice of the performance indicator need to be made according to the particular job at hand and the associated consequences of doing it wrong.

In terms of the precision of the algorithms, it is possible to establish a point of prediction that is ideal. Yet, the error rate of false negatives and/or false positives is the only one that is reduced at this optimal point. According to the findings, the accuracy of the forecast improves with increasing delay. This may be understood by considering the quantity of information that is utilized for learning, which grows with an increasing distance from the prediction point. Even while it may be the most accurate forecast point from a purely technical standpoint, it does not mean that it is the one that should be chosen. The costs that may be avoided owing to an early forecast (by avoiding later process activities), however, are not taken into account, and it is uncertain whether or not the prediction would ultimately be beneficial overall.

The optimization model that has been provided incorporates this trade-off between the degree of precision achieved and the possibility for cost savings into an all- encompassing economic analysis. It gives the ability to establish the various fees for an error as well as the expenses that emerge throughout the execution of the procedure. The ideal thresholds for the classes and the best prediction point may both be calculated in order to optimize the total cost savings; however, the optimal thresholds and prediction point will differ depending on the specific circumstance. The forecast point may shift if the ratio of the costs of the procedure to the costs of errors is taken into consideration. As a result, doing an in-depth analysis of the various cost rates involved in the prospective procedure must come first.

5. CONCLUSION

In this paper, we established a method for determining whether or not Machine Learning approaches may be used as a decision support tool for predictive business process monitoring. This method was developed as part of this paper. The forecast is not applied to specific performance metrics, in contrast to the methodologies that came before it. During runtime, it is used to categorise individual instances of a process. The findings of the experiments indicate that a variety of different machine learning approaches are appropriate for the prediction problem that is connected to business operations. The selection of which classifier to use for a certain undertaking is fundamentally determined by the particular requirements of the aim for which the prediction is being made (e.g. should the classification be interpretable or not). In the research that has been done in the past on this subject, it has not yet been determined whether or not judgments that are based on forecasts are justified in terms of the chance to save costs in comparison to the probable expenses for making incorrect predictions. In this paper, we established a method for determining whether or not Machine Learning approaches may be used as a decision support tool for predictive business process monitoring. This method was developed as part of this paper. The forecast is not applied to specific performance metrics, in contrast to the methodologies that came before it. During runtime, it is used to categorise individual instances of a process. The findings of the experiments indicate that a variety of different machine learning approaches are appropriate for the prediction problem that is connected to business operations. The selection of which classifier to use for a certain undertaking is fundamentally determined by the particular requirements of the aim for which the prediction is being made (e.g. should the classification be interpretable or not). In the research that has been done in the past on this subject, it has not yet been determined whether or not judgments that are based on forecasts are justified in terms of the chance to save costs in comparison to the probable expenses for making incorrect predictions. Another potential expansion would be to not compare the judgment of the algorithm with that of a person, but rather with an ex-post objective result (e.g. whether or not a component exceeded tolerance range in the end or whether a patient diseased). This would be a concession to the reality that the human judgment may not always be the optimal one depending on the circumstances. In the context of business process monitoring, our approach makes it possible for us to assist in evaluating the potential financial impact of projections. Our findings demonstrate that approaches including machine learning are useful for business process management in general and process monitoring in particular. We offer a solid foundation for future study in the aforementioned fields.

6. REFERENCES

1. Al-Turjman, F., Wazan, A. S., & Sali, A. (2018). Anomaly-based predictive maintenance and optimization for IoT-enabled manufacturing systems. IEEE Transactions on Industrial Informatics, 14(1), 5-15.
2. Caron, F., van den Heuvel, W. J., & van Dongen, B. F. (2017). Process mining in healthcare: A literature review. Journal of Data Mining and Knowledge Discovery, 31(3), 1-54.
3. Davenport, T. H. (2013). Process innovation: Reengineering work through information technology. Harvard Business Press.
4. Di Francescomarino, C., Ghidini, C., & Rospocher, M. (2016). Predictive process monitoring methods: Which one suits me best? In International Conference on Business Process Management (pp. 241-257). Springer.
5. Evermann, J., & Rehse, J. R. (2017). Predictive modeling for event logs. Decision Support Systems, 95, 1-12.
6. Leontjeva, A., & Dumas, M. (2019). Predictive process monitoring in operational processes: a case study from the financial sector. Information Systems and e-Business Management, 17(2), 237-277.
7. Song, M., & Eastman, C. M. (2011). BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors. John Wiley & Sons.
8. Tax, N., Verenich, I., La Rosa, M., Dumas, M., & Maggi, F. M. (2017). Predictive business process monitoring with LSTM neural networks. In Proceedings of the International Conference on Advanced Information Systems Engineering (pp. 477-492). Springer.
9. Van der Aalst, W. M. P. (2011). Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer Science & Business Media.
10. van der Aalst, W. M. P., de Medeiros, A. K. A., & Weijters, A. J. M. M. (2005). Process mining with the HeuristicsMiner algorithm. Technological Forecasting and Social Change, 72(2), 131-156.
11. Vanthienen, J., De Weerdt, J., & Baesens, B. (2008). Process mining as a tool for decision support: A case study on incident management. Decision Support Systems, 44(2), 345-360.
12. Weske, M. (2012). Business process management architectures. In Business Process Management (pp. 333-357). Springer.