Exploring the Potentials of Artificial Intelligence Techniques for Business Process Analysis

Sharam Dadashnia, Peter Fettke, Philip Hake, Johannes Lahann, Peter Loos, Sabine Klein, Nijat Mehdiyev, Tim Niesen, Jana-Rebecca Rehse and Manuel Zapp

Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
{sharam.dadashnia, peter.fettke, philip.hake, johannes.lahann, peter.loos, sabine.klein, nijat.mehdiyev, tim.niesen, jana-rebecca.rehse, manuel.zapp}@dfki.de

Abstract. The given BPI Challenge 2017 provides a case study based on a real-life event log from the financial industry. In the present report we explore the applicability of diverse process mining and predictive analytics techniques and tools in the context of a loan application process in order to provide insightful information to the process owner. These techniques include process discovery, process similarity measures, a novel approach for clustering process data to get process fragments, deep learning based approaches for predictive process monitoring, feature correlation and ranking analysis. For each technique, we describe the general approach, experimental settings and a reporting on the results. These results are then used to answer and discuss the specific questions asked in BPI Challenge 2017.

Keywords: process mining, process analysis, artificial intelligence, log clustering, deep learning

1 Introduction

Today's business processes are becoming increasingly complex in both structure and case volume. Due to the advancing digitalization of processes, there are nowadays terabytes of collected process data available, typically in the form of event logs. This data can be extremely valuable for the executing organization, as it allows constantly monitoring, analyzing, and improving the underlying process, enabling reduced cost or improved quality.

Process Mining provides the means to conquer this complexity. It uses the generated event data to discover process models, check their compliance, analyze potential bottlenecks, and suggest improvements [1]. Well-established process mining tools such as Disco, minit, or Celonis are capable of structuring even large process logs and bringing them into a form that is easily understandable by humans. This way, process owners are able to take a first step towards a better understanding and management of their increasing complex processes.

In addition to the existing tools that are specifically tailored towards process logs, there exist various other techniques for data analysis, which can just as well be adapted to be used on process logs. Research fields such as Data Mining or Machine Learning, which can be summarized under “Artificial Intelligence” (AI), provide powerful means for all kinds of data analysis. By applying new and innovative BPM techniques based on current AI research onto process data and combining the results with established process mining tools, we are provided with a plethora of possibilities to make new discoveries and gain valuable insights into all kinds of processes. Demonstrating some of these in the context of the 2017 BPI challenge is the objective of this report. The yearly BPI challenge provides an excellent opportunity for researchers to demonstrate the feasibility of newly developed approaches in a real-life setting by simultaneously helping the involved organizations to better understand and improve their process. This year’s challenge provides us with a log from a Dutch financial institute, describing the loan application process. More specifically, the data consists of two logs, describing all events related to either loan applications initiated by the customer or loan offers made by the bank. For each offer, there is a corresponding application. If an offer exists for an application, it is referenced in the log. What is special about this challenge is that the same process was already considered in the BPI Challenge 2012[1], giving us the opportunity to directly compare the process of today and the process from five years ago. Since the process was first analyzed in 2012, the financial institute has implemented a new workflow system and realized some of the advice that they received during the last challenge. Also, due to the financial crisis, the case volume of the process has risen considerably. According to the statement formulated in this year’s BPI challenge, the financial institute is particularly interested in the following questions:

1. What are the throughput times per part of the process, in particular the difference between the time spent in the company's systems waiting for processing by a user and the time spent waiting on input from the applicant? This is currently unclear.
2. What is the influence on the frequency of incompleteness to the final outcome? The hypothesis here is that if applicants are confronted with more requests for completion, they are more likely to not accept the final offer.
3. How many customers ask for more than one offer (where it matters if these offers are asked for in a single conversation or in multiple conversations)? How does the conversion compare between applicants for whom a single offer is made and applicants for whom multiple offers are made?
4. Any other interesting trends, dependencies

The objective of this report is to provide as much insights into the given data as possible, combining established process mining tools and techniques with innovative approaches to process analysis. These include for example clustering process data to identify subprocesses and training a convolutional neural network to predict following process steps. This report is organized as follows. After this introduction, Section 2 provides a brief overview of the datasets and the analysis tools used throughout this

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

paper. Section 3 reports on the results from a descriptive process analysis. Afterwards, Section 4 shows the comparison of the process models from the dataset_2102 and dataset_2107. Section 5 then reports on the identification of clustered subprocesses and also a clustering approach for business process model similarity. In order to identify certain dependent variables within the dataset, Section 6 provides the results of a Chi-Square-Test. Section 7 describes a novel application of a convolutional neural networks for predictive process monitoring before section 8 concludes the paper.

2 Data Description & Tools

The dataset used in this year's edition of the Business Process Intelligence Challenge (BPIC'17) describes the loan application process in a Dutch Financial Institute and, furthermore, represents the same process that already constituted the dataset for the BPIC'12 five years ago.[2] It comprises two different datasets containing event log data for both the application process and the offer creation processes.

Tab. 1 provides an overview of the individual datasets along with a brief description and some basic statistics. All data files were provided in standard *xes* format. XES is an acronym for eXtensible Event Stream and builds on the XML file format. Thus, it provides clear structure, can be easily parsed and generated and is flexible enough to capture detailed event log data as well as rich additional process information. Compared to plain text event log, like for instance the widely used *.csv* format, XES provides a much richer presentation of process instance data and additional attributes that can be easily processed by standard process mining tools (e. g. ProM, Disco). The dataset contains event log data from the period between 2016/01/01 and 2017/02/01.

Furthermore, for both event logs there exists an additional unique ID, i. e. each event can be uniquely identified not only in its own event log but also between the logs. For the analyses described hereafter, a variety of both commercial and open-source tools have been used as well as software tools developed specifically for reference model analysis at DFKI (German Research Center for Artificial Intelligence)[3]. Regarding descriptive analytics, standard process mining tools have been applied while customized software components were used for predictive analytics using Deep Learning techniques. The following table provides a summary of the software frameworks and tools that have been applied for data preprocessing and analysis.

[2] http://www.win.tue.nl/bpi/doku.php?id=2012:challenge

[3] http://refmod-miner.dfki.de

Tab. 1. Event log files provided in the context of BPIC'17.

Name	Description
Application event log	The event log contains all events related to the loan application process as well as additional information for the individual instances. Within the log, different types of events resulting from different subprocesses or applications can be distinguished: A-type events (marked by the prefix A in the event description) refer to the subprocess of application handling, O-type events describe the subprocess of offer creation and subsequent activities while W-type events refer to workflow activities. In total, the log contains 561,671 events from 31,509 individual loan applications (i. e. process instances or cases). For every instance, 15 additional attributes exist besides the unique case ID, timestamp and event description. Attributes comprise, for example, the requested loan amount (value in the currency Euro), the applicant's credit score (integer rating), the reason the loan was applied for (categorical data), and, the number of terms for an application (integer number).
Offer event log	The offer event log contains all events related to the process of offer creation and handling these results from incoming loan applications. There is a total of 193,849 events recorded in the log, corresponding to 42,995 offers. Besides the case ID, timestamp and event description, there are 14 more attributes in the log, describing for instance the amount that was offered to the applicant and the initial withdrawal amount (both amounts in currency Euro), the number of agreed payback terms (integer value), and the monthly costs (amount in Euro).

Tab. 2. Software frameworks and tools employed for data preprocessing and analysis.

Framework/Tool	Application purpose
Aris	Process visualization and modelling
Disco	Process mining, process discovery, behavior analysis
ProM 5.2	Sequence clustering, data conversion
Python	Data preprocessing, data querying, data conversion
R and RStudio	Data discovery, descriptive analysis, data manipulation, graphs
RefMod-Miner	Model comparison, reference model mining
Tensorflow	Deep Learning
Weka	Machine Learning, regression analysis

3 Descriptive Process Analysis

3.1 Description of the BPI2017 Process

The global loan application and handling process consists of the three subprocesses (i. e. the application, offer, and workflow subprocess) containing a total of 26 activities (cf. Fig. 1). From a high-level perspective, the global process can be described as follows: the process starts with the creation and submission of a new application. After that, some internal preparatory steps are performed to run automatic checks on the application and set up a new workflow to handle the leads. In some cases, additional information is requested from the applicant before the application can be further processed. Afterwards, the complete application is validated, which may result in three outcomes: the application can immediately be declined (e. g. due to formal issues), directly be accepted (called "shortened completion") or be validated in more detail. In the first case, the application will be canceled and the process will be terminated. In the second case, an offer is created, sent to the customer (by mail or online only) and discussed with them on the phone. In the third case, an in-depth analysis is carried out, which encompasses fraud detection and optional requests for possibly incomplete application files. As a result, an offer can be created when the validation is passed or the application can be declined when the validation fails.

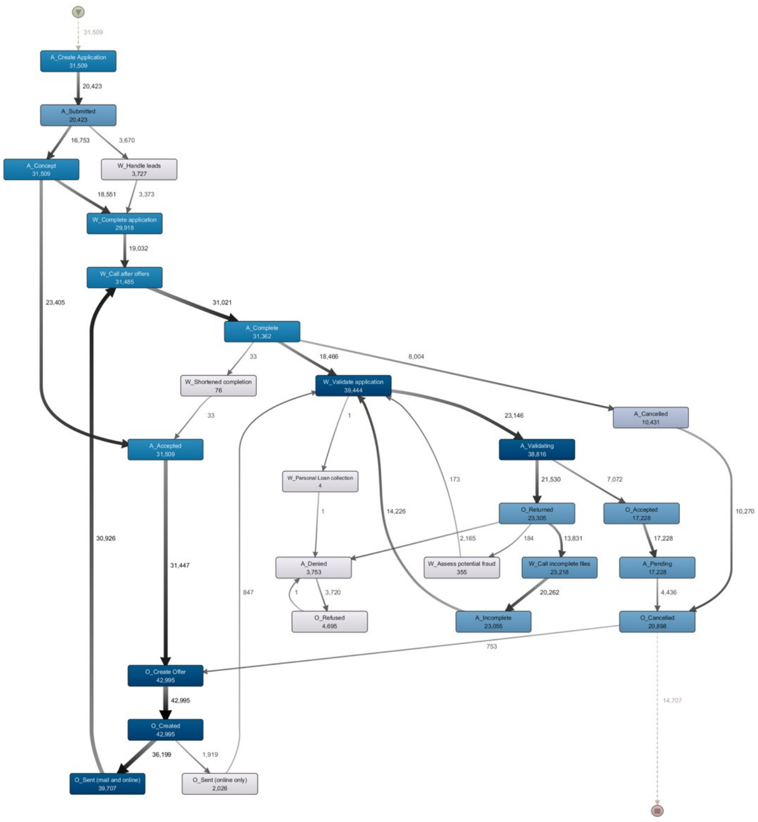


Fig. 1. Process model of the overall process visualized in Disco (parameters: 100% activities, 0% paths).

3.2 Description of the BPI 2012 process

In order to investigate differences between the current dataset and the previous implementation of the processes in 2012, we compare the BPI2017 event logs against the BPI2012 dataset. To lay the foundation for the comparison, the 2012 version of the process is briefly summarized in the following. It consists of 36 activities and is also composed out of three subprocesses (cf. Fig. 2).

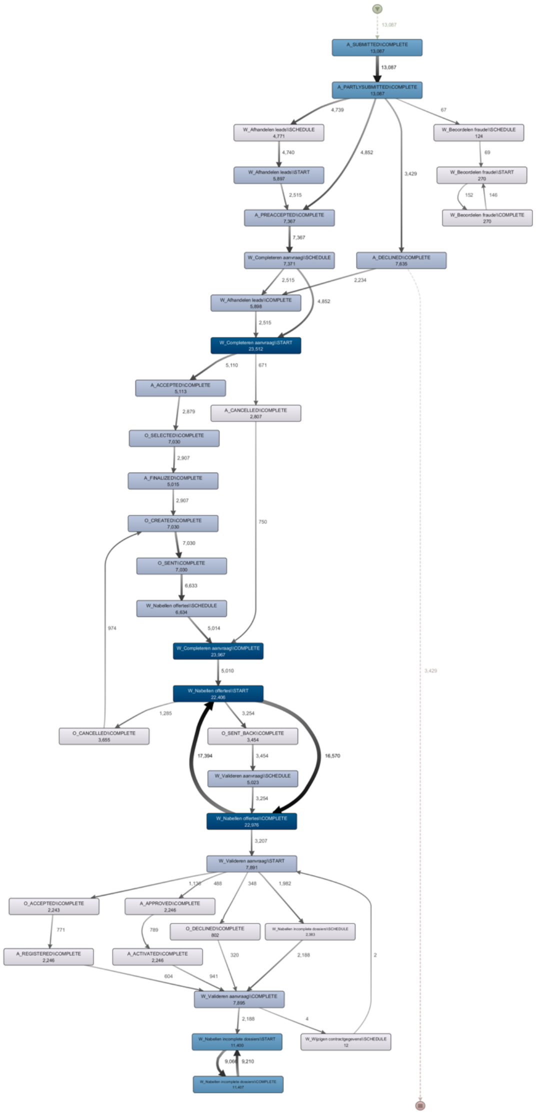

13,087
A_SUBMITTED\\COMPLETE 13,087
13,087
A_PARTLYSUBMITTED\\COMPLETE 13,087
4,739
67
W_Afhandelen leads\\SCHEDULE 4,771
W_Beoordelen fraude\\SCHEDULE 124
4,740
4,852
3,429
69
W_Afhandelen leads\\START 5,897
W_Beoordelen fraude\\START 270
2,515
152
146
A_PREACCEPTED\\COMPLETE 7,367
W_Beoordelen fraude\\COMPLETE 270
7,367
W_Completeren aanvraag\\SCHEDULE 7,371
A_DECLINED\\COMPLETE 7,635
2,515
2,234
W_Afhandelen leads\\COMPLETE 5,898
4,852
2,515
W_Completeren aanvraag\\START 23,512
5,110
671
A_ACCEPTED\\COMPLETE 5,113
2,879
A_CANCELLED\\COMPLETE 2,807
O_SELECTED\\COMPLETE 7,030
2,907
A_FINALIZED\\COMPLETE 5,015
2,907
O_CREATED\\COMPLETE 7,030
750
7,030
O_SENT\\COMPLETE 7,030
6,633
W_Nabellen offertes\\SCHEDULE 6,634
974
5,014
W_Completeren aanvraag\\COMPLETE 23,967
5,010
3,429
W_Nabellen offertes\\START 22,406
1,285
3,254
O_CANCELLED\\COMPLETE 3,655
O_SENT_BACK\\COMPLETE 3,454
17,394
3,454
16,570
W_Valideren aanvraag\\SCHEDULE 5,023
3,254
W_Nabellen offertes\\COMPLETE 22,976
3,207
W_Valideren aanvraag\\START 7,891
1,136
488
348
1,982
O_ACCEPTED\\COMPLETE 2,243
A_APPROVED\\COMPLETE 2,246
771
789
O_DECLINED\\COMPLETE 802
W_Nabellen incomplete dossiers\\SCHEDULE 2,383
A_REGISTERED\\COMPLETE 2,246
A_ACTIVATED\\COMPLETE 2,246
320
2,188
2
604
941
W_Valideren aanvraag\\COMPLETE 7,895
2,188
4
W_Nabellen incomplete dossiers\\START 11,400
W_Wijzigen contractgegevens\\SCHEDULE 12
9,066
9,210
W_Nabellen incomplete dossiers\\COMPLETE 11,407

Fig. 2: Process model of the overall 2012 process visualized in Disco (parameters: 100% activities, 0% paths).

The process starts by the submission of a loan application that is followed by initial automatic checks as well as fraud detection activities. In case of missing information, the applicant is contacted in order to complete the application. Once an application is complete and it is not declined for formal issues or for not passing automatic checks, an offer is created and presented to the applicant. Afterwards, customers are called for discussing the offer. Finally, the application is assessed and a decision on its approval or dismissal is made.

3.3 Process Comparison: Differences and Similarities

Subsequent to the individual descriptive analyses of the BPIC 2012 and BPIC 2017 event logs, a detailed comparison between the two datasets was performed. To get a first impression on their similarities and differences, some basic metrics like the average number of events per case and the medium/average durations of cases were therefore extracted. Tab. 3 summarizes the metrics for both datasets where *dataset_2012* denotes the event log from BPIC 2012 and *dataset_2017* denotes the application event log from this year's BPIC.

Tab. 3. Comparison of process log metrics.

	Dataset 2012	Dataset 2017 (application log)
Events	262,200	561,671
Cases	13,087	31,509
Events per case	20,0	17,8
Activities	36	26
Median case duration	19.4 hrs	19.1 d
Mean case duration	8.6 d	21.9 d
Time period	5.5 months	13 months
Cases per month	2,380	2,420

Number of events per case. The table clearly shows that dataset_2017 contains slightly more than twice as much events compared to dataset_2012 (factor 2,14) while the number of cases increased by a larger factor of 2,4. As a consequence the average number of events declines from 20 in dataset_2012 to about 17,8 in dataset_2017.

This is remarkable since at the same time, the number of "short cases" (containing 6 or less activities) tremendously declined. In dataset_2012, the following two short variants had a large share of the total set of cases: variant 1 (containing 3 events, contributing 26,2% of cases) and variant 2 (containing 6 events, contributing 14,3% of cases) alone had a joint share of 40,5% of all cases. From a business perspective, those variants subsume loan applications that very immediately (variant 1) or after basic checks (variant 2) were declined. Regarding the remaining 59,5% of cases, there are a lot of "long cases" containing much more than 20 events, which results in the average of 20 events per case. To sum up, in dataset_2012 there is a large amount of both "short" and "long" cases regarding the number of events per case. For dataset_2017 the situation is very different: the most frequent variant contains 12 events and only has a share of 11,6% of all cases while the second most frequent variant contains 19 events and has a share of 4,61%. Thus, the share of "short" or "long" cases is much less substantial than in dataset_2012, which means that the number of events is much more evenly distributed over the log. The business perspective again shows that indeed there are much less cases in dataset_2017 where applications were almost immediately declined.

One possible explanation of the observed issue could be a structural or regulatory change in the initial checks within the application process. For instance, the policy on how incoming applications are checked for feasibility and may have changed in such a way that applications are nowadays investigated in more detail – possibly always by employees instead of automatisms – before they can be declined.

Case duration. When comparing on-average case durations between dataset_2012 and dataset_2017, great differences become evident. While the mean case duration varies between 8,5 days and 21,9 days (dataset_2012 vs. dataset_2017) – which corresponds to a factor of 2,6 – the difference in the median case duration is much higher: 19.4 hours compared to 19.1 days (factor 23,6). Fig. 3 and 4 also demonstrate this issue by showing the frequency distribution of case durations in histogram plots.

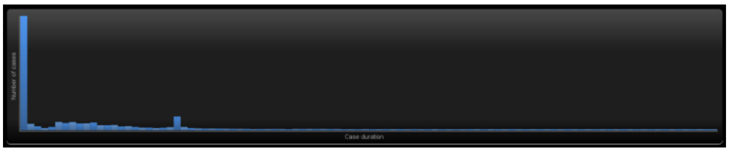


Fig. 3. Frequency Distribution of case durations in dataset_2012.

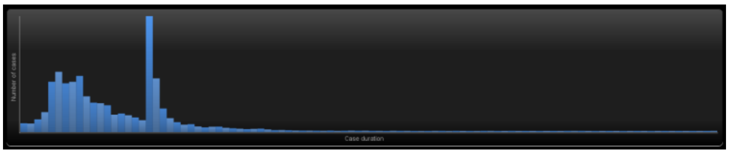


Fig. 4. Frequency Distribution of case duration in dataset_2017.

In Fig. 3, an accumulation of short case durations (high bar on the left side of the histogram) followed by several minor amplitudes can be observed. Long-running cases barely exist. On the other hand, Fig. 4 shows a very different picture: although the histogram also shows an accumulation within the first third of the x-axis, values are distributed in a more heterogeneous way. Overall, there are many cases with higher duration times compared to dataset_2012, which in turn explains the large difference in median case durations and the (relatively) small difference in mean case durations.

Related to the analysis on the number of events per case, a similar explanation for this behavior seems plausible. As already mentioned, dataset_2017 does not contain many cases where an application is immediately declined by automatic checks running the application system. Instead, there seem to be very few cases where a decision on whether to approve or decline an application is determined so quickly but cases tend to be more time-consuming on average. The left part of the frequency distribution in Fig. 4 also resembles the shape of a normal distribution, indicating that there are some cases with short durations, increasingly more cases with longer durations and then again fewer cases with even longer durations. There is one additional peak in the histogram which also yields the highest frequency of all durations but very quickly drops back to lower levels and eventually approaches the x-axis. From a business perspective, this peak is interesting since it clearly marks a relatively large set of long-running cases that are likely to be connected in some way.

Conspicuous process patterns. Furthermore, as we generate the process graph from both datasets and compare the process flow to get some insights regarding the most time-consuming process fragments, we see some differences between the dataset_2012 and the dataset_2017. As we already mentioned, the process step to cancel an application process took much more time in the process of 2017 than in the process from 2012. In addition, we can see some inefficiency within the process of the dataset_2012. For example, the mean duration in the process step from assessing the application to filling in the information for an application took about 30.7 days to execute the process step (cf. Fig. 5). This could mean, that in the past the check of the application took much more time, because there were a lot of manual process steps as the application process was not digitalized and not supported by an online form.

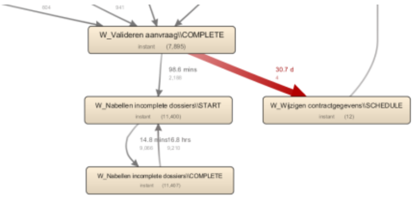

604
941
W_Valideren aanvraag\\COMPLETE
instant (7,895)
98.6 mins
2,188
30.7 d
4
W_Nabellen incomplete dossiers\\START
instant (11,400)
W_Wijzigen contractgegevens\\SCHEDULE
instant (12)
14.8 mins
9,066
16.8 hrs
9,210
W_Nabellen incomplete dossiers\\COMPLETE
instant (11,407)

Fig. 5. Dataset_2012, mean duration, 100% activities, 0% paths.

As can be seen in Fig. 6, the maximum of the duration in this process step was 14.7 weeks. Regarding the customer satisfaction this could be a very misleading process behavior, if we assume that the customer is waiting for a feedback on the application. On the other hand, the process execution of the 2017 dataset contains some very time-consuming process steps like the process of canceling an application.

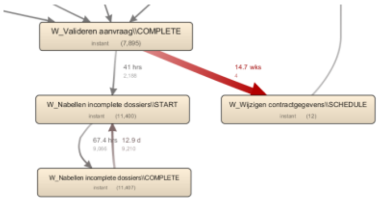

W_Valideren aanvraag\\COMPLETE
instant (7,895)
41 hrs
2,188
14.7 wks
4
W_Nabellen incomplete dossiers\\START
instant (11,400)
W_Wijzigen contractgegevens\\SCHEDULE
instant (12)
67.4 hrs
9,066
12.9 d
9,210
W_Nabellen incomplete dossiers\\COMPLETE
instant (11,407)

Fig. 6. Dataset 2012 max duration, 100% activities, 0% paths.

In Fig. 7 the mean duration of the cancellation process step took about 27 days. Furthermore, this process step is executed in total 8,004 times. This could be also a problem regarding the customer satisfaction, because this time-consuming step leads also to a delay of the customer communication.

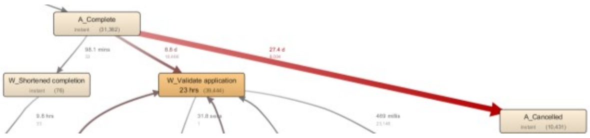


Fig. 7. Dataset_2017, mean duration, 100% activities, 0% paths

4 Compare the Process Models form 2012 and 2017

4.1 General Approach

The objective of this chapter is to compare the given event log with the event log provided in the BPI Challenge 2012[4], which described the same process five years prior. By using state-of-the-art process similarity and comparison techniques, we get the chance to point out similarities and differences between the processes and confirm or reject the usefulness of implemented changes. Therefore, we mine a process model from each log, define a matching between them, and use different similarity measures to assess their differences and commonalities.

The basis for each comparison is a manual matching between the process activities. To increase its validity, matchings were made by two researchers independently. The results were discussed and then merged into the final matching. In order to achieve the best results, we had to rely on a manual matching between the activities instead of using one of the many automated matching approaches [2] implemented in the RefMod-Miner. The main reason is that some events are equivalent but would not be matched automatically without having any contextual information, e. g. the nodes “A_Concept” and “A_PREACCEPTED”. On the other hand, there are activities which would be matched because of their similarity but describe different events, e. g. “A_Accepted” and “A_PREACCEPTED”. Based on the mined models and this matching, we assess their similarity by calculating the percentage of common nodes, percentage of common nodes and edges, graph edit distance, causal footprints as described in [3], and model similarity based on behavioral profiles [4]. The first two measures are rather straightforward, when taking the defined matching into account. The graph edit distance is a heuristic measure estimating the editing steps that are necessary to transform one process model into the other. Causal footprints denote precedence relations between activities, whereas behavioral profiles analyze the relations between activities on a more concrete level.

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

4.2 Experimental Settings

First of all, since most similarity measures are defined on process models, we discovered a process model for both logs, using the state-of-the-art Inductive Miner algorithm (IMi), with standard settings [5]. Since the RefMod-Miner does not handle Petri Nets, the results were manually transformed into Event-driven Process Chains (EPC) using ARIS[5]. For the creation of the mapping as well as the computation of the similarity measures, we used RefMod-Miner functionalities, such as a Mapping Editor and several implementations of similarity measures.

4.3 Results

Fig. 8 shows the discovered process models for each log, 2012 on the left and 2017 on the right. The matching between the activities is displayed in Tab. 4. The calculated similarity measures are listed in the following Tab. 5.

Tab. 4: Matching between Activities of BPI 2017 and BPI 2012

Activity 2017	Activity 2012	Activity 2017	Activity 2012
A_CreateApplication	---	O_Accepted	O_ACCEPTED
A_Submitted	A_SUBMITTED	A_Pending	---
A_Concept	A_PREACCEPTED	A_Denied	A_DECLINED
W_Complete application	W_Completeren aanvraag	W_Personal Loan collection	---
A_Accepted	A_ACCEPTED	O_Cancelled	O_CANCELLED
O_Create Offer	O_SELECTED	W_Handle leads	W_Afhandelen leads
O_Created	O_CREATED	A_Cancelled	A_CANCELLED
O_Sent (mail and online)	O_SENT	W_Assess potential fraud	W_Beoordelen fraude
W_Call after offers	W_Nabellen offertes	O_Sent (online only)	O_SENT
A_Complete	A_FINALIZED	O_Refused	O_DECLINED
W_Validate application	W_Valideren aanvraag	W_Shortened completion	---
A_Validating	---	---	A_REGISTERED
O_Returned	O_SENT_BACK	---	A_APPROVED
W_Call incomplete files	W_Nabellen incomplete dossiers	---	A_PARTLYSUBMITTED
A_Incomplete	---	---	A_ACTIVATED

[5] http://www2.softwareag.com/corporate/products/bis/default.aspx

Tab. 5. Similarity measures for BPI 2017 and BPI 2012

Similarity Measure	**Value**
Percentage of common nodes	80%
Percentage of common nodes and edges	49%
Graph edit distance	64%
Causal footprints	87%
Behavioral profile similarity	34%

The first two similarity measures describe the correlation of nodes and edges in the two process models. While the percentage of common nodes is relatively high, the percentage of common nodes and edges is only about 50%. This means that the models share a lot of identical or similar nodes but the process structure has considerably changed since 2012. This finding also explains the average value for the graph edit distance as more edges than nodes had to be altered to get from one model to the other. The high number of causal footprints can be explained by the fact, that both process models also have a high number of loops and back-edges. Therefore, a lot of nodes are not followed by a single but by multiple other nodes which results in a higher number of causal dependencies compared to a sequential model. The behavioral profiles, on the other hand, have a low similarity value, indicating a lower degree of conformance regarding the individual activity relations in the model. This can also be explained by the changed structure.

The calculated similarity measures indicate that the process content, i.e. the executed activities, has not significantly changed since 2012, but the process structure has. This correlates with the information that the process has been re-implemented using a new workflow system, realizing advice from the previous challenge. By going deeper into the process models, we can identify and assess some of the implemented changes. First of all, the 2017 model has 26 activities, whereas the 2012 model has 24. As we see from the matching, 20 activities can be matched. Four activities (A_REGISTERED, A_APPROVED, A_PARTLYSUBMITTED, A_ACTIVATED) have been removed from the process, most likely because they were obsolete or duplicate. On the other hand, six new activities were included in the process. Two of them (W_Personal loan collection, W_Shortened completion) are workflow items, i.e. manual process steps, while four are application states (A_Create application, A_Validating, A_Incomplete, A_Pending). Each state represents a certain application condition, where it is necessary to take action, which is why they are all immediately followed by another state or a workflow item.

Another important difference is the position of some activities in the process. For example, in the 2012 model, applications are checked for fraud immediately after submission, with a mean duration of about ten minutes. This step is executed in about 0.8% of all cases, but has to be repeated up to nine times. In the 2017 model, this check appears much later, only after an offer is rejected, but it has a mean duration of three days. It is executed in about 1.3% of all cases, but only has to be repeated three times at most. This indicates that the nature of the check has changed. It is now only executed, once there are substantial clues. Due to the position later in the process, more evidence for a potential fraud can be considered. Also, the execution times and repetitions indicate that the step is now executed manually and more thorough.

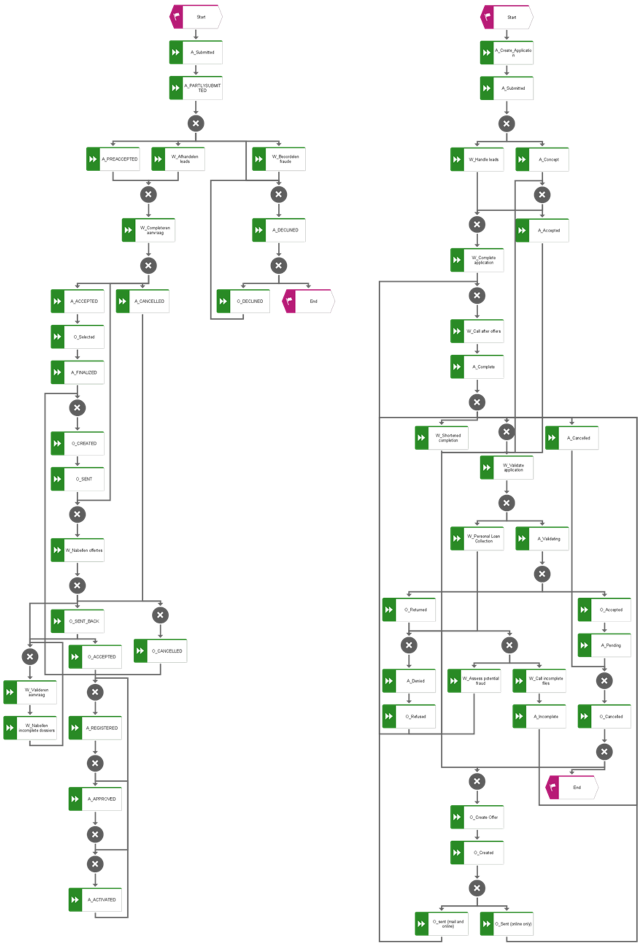


Fig. 8. Mined models for the process logs of 2012 (left) and 2017 (right)

5 Identification and Analysis of Subprocesses

5.1 General Approach

In order to gain a deeper understanding of the business process underlying the challenge data set, we analyze it using the RefMod-Miner research prototype. This prototype is designed to enable process model analysis and reference model mining and, for this purpose, contains a variety of different helpful techniques. In particular, we use the RefMod-Miner to apply a clustering approach in order to mine reference model components from the log. Reference model components are small and structured process fragments that can be identified within a log. The idea is that particularly large process logs, such as the challenge data set, are often too complex to mine a single meaningful process model. Especially non-robust mining approaches will result in highly unstructured "Spaghetti-models", which are hard to read and offer little to no analytical value. This problem is often addressed by dividing the log horizontally, clustering cases into subsets based on similarity [6]. Here, we use a different approach and divide the log vertically. By separating it into shorter sequences, we are able to mine smaller and more structured subprocesses, which allow to analyze the process on a higher level of detail.

To identify the components, we also employ a clustering approach. However, the clustering is applied to the set of activities contained in the log instead of the set of traces. Activities are clustered based on their distance in the log, following the idea that activities which often appear in close proximity to each other form a logical unit and thus follow a somewhat clear structure. For determining the distance between two activities, we first select the set of traces containing both activities at least once. For each trace, the distance measure is calculated by counting the number of steps between the two activities, dividing it by the length of the trace to get a normalized value and deducting the result from 1 to get a similarity value. If the activities appear multiple times within one trace, the average distance is calculated. The distance between two activities is then defined as the normalized arithmetic mean across all traces. If two activities do not have a common trace, the similarity value is 0. Distance values are computed for all pairs of activities, resulting in a similarity matrix. This matrix is used as input for a clustering algorithm, resulting in several activity clusters. For each cluster, we mine a reference model component containing only the specified activities. Since these components are smaller, we are able to inspect them individually and more closely, while avoiding "Spaghetti-like" process models. This enables us to analyze the throughput times for each process part individually.

5.2 Experimental Settings

For clustering the activities, we use the k-means clustering algorithm, using the Hartigan-Wong variant and not specifying the expected number of clusters [7]. An efficient implementation is available in R and called directly from the RefMod-Miner. The reference model components are mined using the Disco filters, which remove all other activities from the log, reducing it to the specified cluster, for which a process model can be mined using the regular Disco functionality.

5.3 Results

Fig. 9 shows the result of clustering the activities of the challenge log based on their distance. The colors indicate similarity values; green stands for high, red for low values of similarity between activities. The values are arranged in a matrix and ordered by the result of the clustering, resulting in a heatmap. The k-means algorithm returned seven clusters for the given set of activities, which are fairly easily discernible in the heatmap.

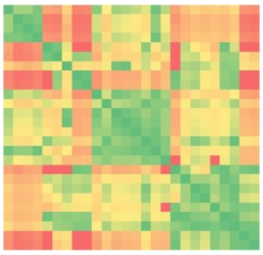


Fig. 9. Heatmap obtained from clustering activities based on distance

Cluster 1 consists of four activities. They form the first subprocess, where a loan application is created (A_Create Application), submitted (A_Submitted), and pre-checked automatically (A_Concept, W_Handle Lead). As we can see from the mined component in Fig. 10, there is a clear structure among the activities with little variance. *Cluster 2* contains three activities, namely A_Cancelled, O_Cancelled, and O_Sent (online only). One can see from the heatmap, that the latter activity has a lower distance similarity, so this cluster assignment may not be optimal. This is also visible from the mined component in Fig. 10, which contains a lot of variance and no clear structure between the activities, although typically A_Cancelled precedes O_Cancelled. This component describes the part of the process where either applications or offers are cancelled, leading to the cancellation of related items.

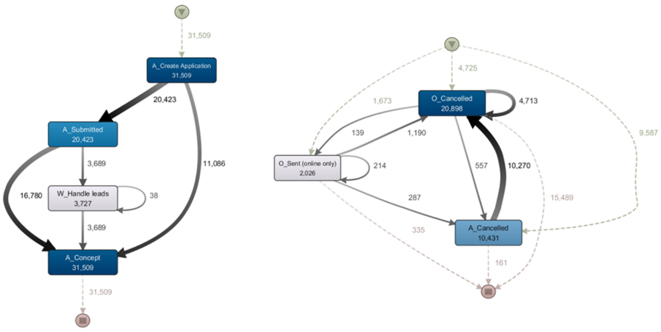


Fig. 10. Subprocesses mined for Cluster 1 and Cluster 2

Cluster 3 is also a small subprocess with a clear and simple structure, as seen on the left of Fig. 11. It describes that, after an offer is accepted (O_Accepted), the corresponding application is set as pending (A_Pending), meaning that the accepted offer is waiting for confirmation by the bank in order to be confirmed and closed. *Cluster 4* is the largest of the subprocesses with eight activities in total, depicted on the left of Fig. 11. It describes the core part of the process, where after an application is completed (W_Complete application) and initially accepted (A_Accepted), an offer is created by the bank (O_Create offer, O_Created) and sent to the customer (O_Sent (mail and online)). If a customer doesn't answer, the offer is called after (W_Call after offers), until it is complete (A_Complete). In a few cases, the completion can be shortened (W_Shortened completion). These activities are usually executed in a defined order with only one possible loop, when a new order is created after an application is completed. Cluster 5, shown in the middle of Fig. 11 is the smallest of all clusters, containing only one activity (W_Personal loan collection). However, since this activity is only executed four times in total, this assignment makes sense.

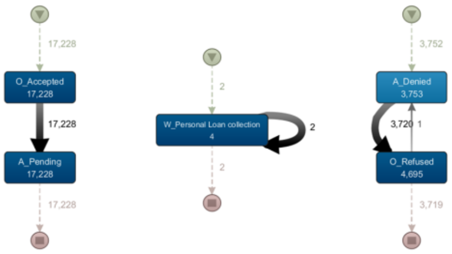


Fig. 11. Subprocesses mined for clusters 3 (left), 5 (middle), and 7 (right)

Cluster 6 describes the subprocess for completing an incomplete application, which is also clearly structured. After validating an application manually and automatically (W_Validate application, A_Validating), the offer is returned (O_Returned), assumedly due to incomplete files, which are then called for (W_Call incomplete files). A is considered incomplete (A_Incomplete) and the subprocess is started over. In few cases, the application is checked for potential fraud after the offer is returned (W_Assess potential fraud). *Cluster 7* is another small cluster with two activities, which describes the final subprocess. If the loan is denied (A_Denied), the offer is refused.

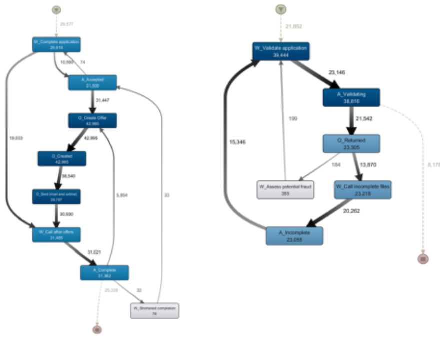


Fig. 12. Subprocesses mined for clusters 4 (left) and 6 (right)

This identification of reference model components allows for two insights. First of all, it is possible to identify subprocesses by simply clustering activities based on their distance. The subprocesses not only exhibit a clearer structure in itself than in the complete model, they can also easily be distinguished by their function in the complete process. Therefore, it is possible to analyze throughput times for each subprocess individually, allowing for better insights of bottlenecks and other difficulties. For each subprocess, Tab. 6 shows the minimum, maximum, median, and mean duration and gives a first hint towards the more or less time-intensive process parts. Unsurprisingly, the smaller and more automated subprocesses mined from clusters 3, 5, and 7 take very little time and can be neglected. Subprocess 1 is also not a top priority, as it has reasonable mean and median durations, although we could look into what caused a five-day delay in the maximum cases. Subprocess 2 appears to be quite volatile in its durations, which is most likely caused by the high variability in the process. The high difference between mean and median indicate that there are only few outliers with a really long duration, giving this subprocess a lower priority as well. This leaves time-intensive subprocesses 4 and 6 to be analyzed.

Inspecting the activities of subprocess 4, we see that W_Complete application and W_Call after offers take the most time. For subprocess 6, W_Assess potential fraud has the highest median, but is executed very infrequently. Besides that, W_Validate application and W_Call incomplete files have the highest durations.

Tab. 6. Analysis of durations per subprocess

Subprocess	**Durations**			
	Minimum	**Maximum**	**Median**	**Mean**
Cluster 1	1ms	5d 33m	49s	43m 54s
Cluster 2	0ms	134d 1h	27ms	52h 42m
Cluster 3	2ms	2s 258ms	4ms	8ms
Cluster 4	26s 381ms	129d 4h	2h 12m	71h 48m
Cluster 5	2s 841ms	6s 390ms	4s 600ms	4s 600ms
Cluster 6	0ms	167d 20h	70h	5d 3h
Cluster 7	0ms	1m 39s	35ms	75ms

6 Analysis of Interdependence between Process Attributes and Outcome

6.1 General Approach

The questions 2 and 3 of the BPI Challenge 2017 require investigating various relationships between the process characteristics and application process outcomes. Particularly, the process owners ask the participants to investigate the association between:

- incompleteness and the outcome of the application process (Question 2)
- the number of offers and the outcome of application process (Question 3)

In order to address both questions, we provide a descriptive analysis of the required variables and conduct non-parametric tests to check whether the association between them is statistically significant. Since we aim to answer the question if two categorical variables are interrelated based on distribution of the cases, we adopt a popular technique, the Chi-square (χ^2) test of independence also known as Pearson Chi-square test [8]. The main advantage of a Chi-square (χ^2) test over existing alternatives is that it cannot only identify the relationship between variables statistically but also provides information about the source and direction of the detected association [9]. There are at least four approaches available to investigate further a statistically significant omnibus chi-square test result: calculating residuals, comparing cells, ransacking, and partitioning [9]. In the present report we use the "comparing cell" approach to further investigate the details of the association [10], [11].

Furthermore, due to its non-parametric nature, the Chi-square (χ^2) technique is suitable for the analysis of the underlying BPI Challenge 2017 data, since the sample size of the study groups is differing and cannot be handled by parametric test methods, which require equal or approximately equal size.

The Chi-square ($\chi2$) starts with the statement of the hypothesis based on the formulated business question. The null-hypothesis suggests that there is no relationship between the variables. If the test results reject the null-hypothesis then there is a relationship between the variables. The formula for calculating the Chi-square ($\chi2$) statistic is as follows:

$$\sum \chi^2_{(i-j)} = (O-E)^2 / E \qquad (1)$$

O is the observed cell total, E is the expected cell totals, i-j represents all the cells from the first cell (i) to the last cell (j). The expected values E is calculated as follows:

$$E = M_R * M_C / n \qquad (2)$$

M_R is row marginal for the cell; M_C is the column marginal for the cell; n is the total sample size. After calculating the χ^2-value, we calculate the degrees of freedom by using the formula:

$$D_f = (\text{Number of rows} - 1) * (\text{Number of columns} - 1) \qquad (3)$$

At the final stage, the Chi-square (χ^2) distribution table is used to match the degrees of freedom and pre-defined probability level to find out the corresponding probability at the identified χ^2-value. If it is smaller to the accepted significance level (which is 0.05 in most studies), the null-hypothesis is rejected, otherwise it is accepted. In the following subsections, we provide a descriptive analysis of the test variables which will be followed by the statistical tests.

6.2 Experimental Settings

Distribution of Process Outcomes. Since in our test one of the nominal variables is the process outcome, we provide detailed information about the possible endpoints. In the BPI Challenge 2017 dataset, the application processes have three possible outcomes, A_Pending (positive), A_Cancelled (negative) and A_Denied (negative). There are also some application process instances, which do not have information about the final status. Those were categorized under "Unresolved". Tab. 7 provides an overview to the definition of these process endpoints.

Tab. 7. Description of potential process endpoints

	Definition
A_Pending	If all documents are received and the assessment is **positive,** the loan is final and the customer is payed.
A_Cancelled	If the customer never sends in his documents or calls to tell he doesn't need the loan, the application is cancelled.
A_Denied	If somewhere in the process the loan cannot be offered to the customer, because the application doesn't fit the acceptance criteria, the application is declined, which results in the status 'denied'.
Unresolved	The process has not ended yet.

Fig. 13 provides an overview to the distribution of the application process outcomes in the BPI Challenge 2017 data. As depicted in Fig. 13, out of 31,509 total applications 17,228 applications ended with the event “A_Pending”. This information suggests that slightly more than half of the applications (55%) ended with the desired outcome where the customer was paid. 33% of customers (10,431) did not accept/reply the loan offered by the bank so the bank had to cancel the applications (“A_Cancelled”). 12% of the applications by customers did not fit the acceptance criteria of the bank. The bank denied granting the loan (“A_Denied”). Less than 1% of the applications have not yet ended so we don’t have any information about the final outcome (“Unresolved”).

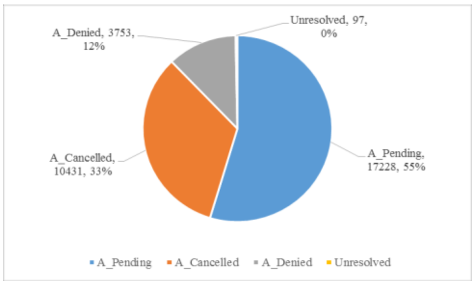
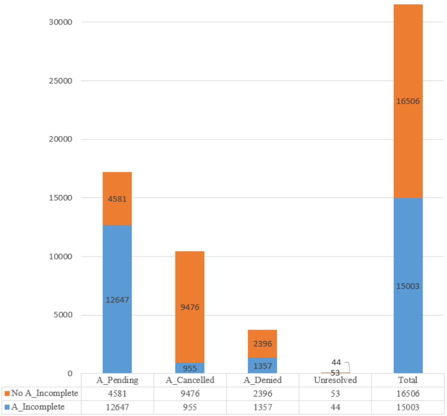


Fig. 13. Outcome Distribution of the Application Processes

Incompleteness in Application Processes. Before answering question 2 about the influence of incompleteness to the final process outcomes, we provide some descriptive information about their associative distribution. The BPIC 2017 forum manager defines the term “incompleteness” in the BPI Challenge 2017 forum as follows: “Incompleteness means how many times an application gets the status 'incomplete'”.

Therefore, in order to test the relationship between incompleteness and process outcomes, we filter the BPI Challenge 2017 data to find out how many applications have at least one “A_Incomplete” status and what proportion doesn’t have any “A_Incomplete” status at all. We have identified that 15,003 unique applications (48% of total 31,509) have at least one “A_Incomplete” status, whereas in 16,506 (52%) applications this status was not observed. Moreover, by using the join functions we identify the distribution of existence “A_Incomplete” and absence of

“A_Incomplete” in terms of each individual process outcomes, namely “A_Pending”, “A_Cancelled”, “A_Denied” and “Unresolved”. Fig. 14 provides a detailed overview to this distribution.



	A_Pending	A_Cancelled	A_Denied	Unresolved	Total
🟧 No A_Incomplete	4581	9476	2396	53	16506
🟦 A_Incomplete	12647	955	1357	44	15003

Fig. 14. Distribution of existence and absence of “A_Incomplete” status in terms of process outcomes, “A_Pending”, “A_Cancelled”, “A_Denied” and “Unresloved”

From Fig. 14 we can visually inspect the relationship between incompleteness and application process outcome. 73% of the applications (12,647 out of 17,228) with the endpoint “A_Pending” have at least one “A_Incomplete” event whereas this number is only 9% for the applications in which customers did not accept the loan offer (“A_Cancelled”). 955 application processes out of total 10,431 processes with the outcome “A_Cancelled” have at least one “A_Incomplete” status whereas 9,476 of them have no incompleteness. Among the application processes with outcome “A_Denied”, 2,396 unique applications have at least one incompleteness, whereas 1,357 are free of incompleteness. For unresolved cases, the distribution is balanced. First impressions from the visual analysis suggest that there is a positive relationship between existence of incompleteness and positive process outcomes. In the next section, we will investigate the relationship with statistical tests.

Single Offer and Multiple Offers. The question 3 requires to conduct both descriptive statistics about the applications where single and multiple offers are required by the customers or offered by the bank and statistical analysis between the number of offers and process outcomes. Fig. 15 provides an insight to each application which has single offer, two offers, three offers, four offers and five and more offers. From this figure, we can easily infer that the majority of applications receive the single offers. 72% of the applications contain only a single offer. The decreasing trend is observed for multiple offers. The number of applications decreases, when the number of offers increases.

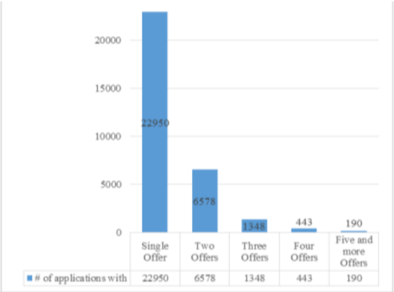


Fig. 15. The number of unique applications with one, two, three, four, five or more offers.

For offers, we also conducted an analysis by matching its categories (single vs. multiple offers) to the application process outcomes. As depicted in Fig. 16 almost 30% of the applications which ended positively (“A_Pending”) – 5,050 out of 17,228 – have multiple offers attached. This number is about 24% for applications which ended with negative outcomes, both for “A_Cancelled” and “A_Denied”. From descriptive analysis we can propose that applications with multiple offers tend to end with positive outcome. However, in order to check the validity of this hypothesis we conducted the non-parametric test which should check whether the association between the number of offers and outcome of applications is statistically significant.

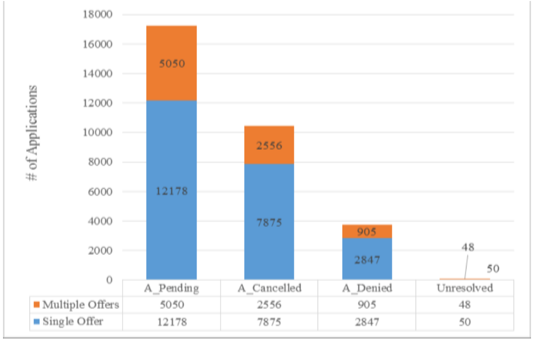


Fig. 16. Distribution of single offers and multiple offers in terms of process outcomes, “A_Pending”, “A_Cancelled”, “A_Denied” and “Unresloved”

6.3 Results

Association between Incompleteness and Process Outcomes. We begin interpreting the relationship between the variables by discussing the overall chi-square tests result which should suggest that whether there is an association between incompleteness and application process outcomes. The null hypotheses in our case states that the incompleteness *doesn’t have* any association with the outcome of the application process which also means that the variables are independent. The alternative hypothesis in contrast suggests that that the information about incompleteness can help us to predict the process outcomes:

- H0: Incompleteness and Process Outcome are independent.
- Ha: Incompleteness and Process are not independent.

The main purpose of the Chi-square ($\chi2$) analysis is to examine whether the null hypothesis is accepted or rejected. In our analysis we defined the significance level as 0.05 as in the state-of-the-art applications. By using the Chi-square ($\chi2$) test for independence we calculated the expected frequency counts, degrees of freedom and chi-square test statistics. Based on the values of the latter two we compute the p-value. The obtained Chi-square ($\chi2$) statistic is 10,978.9316. The p-value is < 0.00001. Since the result is significant at $p < 0.05$, this confirms that there is a statistically significant relationship between the incompleteness and the application process outcomes. The null-hypothesis is rejected. We reveal that both variables are not independent.

Tab. 8. Observed values, expected values in () and cell Chi-square ($\chi2$) values in []

	A_Incomplete	**No A_Incomplete**	**Total**
A_Pending	12,647 (8,203.11) [2407.40]	4,581 (9,024.89) [2,188.19]	17,228
A_Cancelled	955 (4,966.72) [3240.34]	9,476 (5,464.28) [2,945.29]	10,431
A_Denied	1,357 (1,786.99) [103.47]	2,396 (1,966.01) [94.04]	3,753
Unresolved	44 (46.19) [0.10]	53 (50.81) [0.09]	97
Column Total	15,003	16,506	31,509

The results of the conducted statistical analysis revealed the dependency between variables, but we still have to investigate the results in the cell level in order to figure out whether the direction (causality) of the assumption by process owners is right or not. As mentioned above, the process owners suggest that the existence of incompleteness activities leads to negative results.

The results presented in Tab. 8 suggest that the number of observed applications with positive outcome, "A_Pending", which contains at least one "A_Incomplete" activity is 12,647. However, the expected value for this category was only 8,203. The $\chi2$ is 2,407. This implies that the number of applications with positive outcomes ("A_Pending") containing the incompleteness (at least one "A_Incomplete") is significantly greater than expected. In other words, the existence of the incompleteness increases the chance that the application will have positive outcome, "A_Pending".

The second cell also obtains a high χ^2-value with 2,188. This cell provides an overview to each application with positive outcome and without incompleteness. However, a closer look suggests that the number of observed cases (4,581) was significantly lower than the expected value (9,024). This result suggests that a significantly lower number of applications reached the positive outcome ("A_Pending") when there was no incompleteness. In other words, the absence of incompleteness reduces the chance of applications to get positive outcome, "A_Pending". We can make the similar cell-based analysis for the processes with the

"A_Cancelled" outcomes. The results suggest that the number of observed cases for "A_Cancelled" with "A_Incomplete" was significantly lower than the expected values (955 vs. 4,966). At the same time the number of the observed cases "A_Cancelled" without "A_Incomplete" is significantly higher than it was expected. This result suggests that a significantly lower number of applications reached the negative outcome ("A_Cancelled") when there was incompleteness. Or formulated differently, the existence of incompleteness decreases the chances to get negative outcome significantly. The results for the applications with "A_Denied" outcome are similar to the ones with "A_Cancelled". Statistically significantly lower number of applications ended with the "A_Denied" where there was an incompleteness. Also, significantly higher number of applications ended with outcome "A_Denied" where there was no incompleteness. In summary, the analysis of the individual cells of the conducted non-parametric test suggests that the existence of the status "A_Incomplete" increases the chances that the applications will end up with positive outcome, "A_Pending" and decreases the chances the application will end up with one of negative outcomes, "A_Cancelled" or "A_Denied". On these grounds, we can argue that the hypothesis by process owners about the relationship between incompleteness and application process outcomes is wrong.

Association between Number of Offers and Process Outcomes. Similarly, we conduct another Chi-square ($\chi2$) test to check whether there is an association between number of offers and application process outcomes. The null hypotheses in that case states that the number of offers *doesn't* have any association with the outcome of the application process and the alternative hypothesis suggests that the information about the number of offers can be used to predict the process outcomes:

- H0: Number of Offers and Process Outcome are independent.
- Ha: Number of Offers and Process Outcome are not independent.

Again, we define the significance level as 0.05. The results of the Chi-square ($\chi2$) are shown in Tab. 9. In this case, the Chi-square ($\chi2$) statistic is 118.6509, the p-value is < 0.00001 and the result is significant at $p < 0.05$. So, we can confirm that there is also a statistically significant relationship between the number of offers and the application process outcomes. Therefore, the null-hypothesis is rejected because process outcome is not independent of number of offers.

Tab. 9. Observed values, expected values in () and cell Chi-square ($\chi2$) values in []

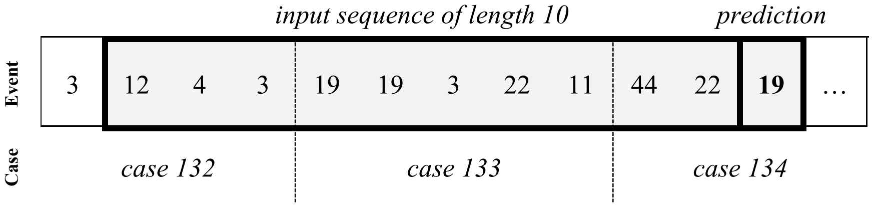
	Single Offer	**Multiple Offers**	**Total**
A_Pending	12,178 (12548.24) [10.92]	5050 (4679.76) [29.29]	17228
A_Cancelled	7,875 (7597.56) [10.13]	2556 (2833.44) [27.17]	10431
A_Denied	2847 (2732.82) [4.77]	905 (1019.18) [12.79]	3752
Unresolved	50 (71.38) [6.40]	48 (26.62) [17.17]	98
Column Total	22950	8559	31509

In order to investigate the details of the association between the number of offers and the process outcomes, we conduct the cell comparison as well. The results presented in Tab. 9 suggest that the number of actual applications with positive outcome ("A_Pending") which contains "Single Offer" is 12,178. However, the expected value for this category was more than observed ones namely 12,548. The Chi-square (χ2) is 10.92. This result suggests that number of applications with positive outcome ("A_Pending") containing "Single offers" is significantly lower than expected. In other words, making single offers reduces the chances to end up with positive outcomes. In contrast the information presented in the second cell suggests that the number of applications with the positive outcome, "A_Pending", which contains "Multiple Offers" is significantly higher than expected (5,050 vs. 4,679). Summarizing, the analysis of these two cells suggest that the applications with multiple offers tend to end up with positive results. Let's now analyze the cells of negative outcomes. The results suggest that the number of actual applications with process outcome ("A_Cancelled") which contains only "Single Offer" is 7,875. However, the expected value for this category was more than observed ones, 7,597. This result implies that the applications with single offers tend to end up with "A_Cancelled" more than expected. In contrast the number of applications with the process outcome "A_Cancelled" which contains "Multiple Offers" is significantly lower than expected (5,050 vs. 4,679). This in turn suggests that making multiple offers reduces the chances to get negative outcomes. The same trend as in "A_Cancelled" can be observed for the second negative outcome category, "A_Denied". Making multiple offers reduces the probability ending up with the process outcome "A_Denied". Whereas, sticking to single offers increases the odds that the application process will have the outcome "A_Denied". In summary, the analysis of the individual cells of the conducted non-parametric test suggests that making multiple offers increase the chances that the applications will end up with positive outcome, "A_Pending" and at the same time decreases the chances the application will end up with one of negative outcomes, "A_Cancelled" or "A_Denied".

7 Process Prediction Using Convolutional Neural Networks

7.1 General Approach

Deep Learning has recently been used to predict process activities in business process execution. In [12, 13] the authors employ Long Short-Term Memory (LSTM) Neural Networks, which are trained using sequences of process execution steps. The evaluated performance of 0.76 accuracy on the 2012 BPI challenge dataset shows that deep learning already yields remarkable results in predicting process events. However, training and validating their neural network requires process sequences of fixed length.



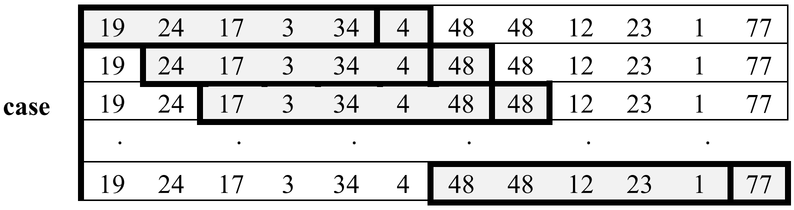
Event	3	12	4	3	19	19	3	22	11	44	22	**19**	…

Fig. 17. Concatenation of Different Process Cases to Generate Training Input.

In [12] the authors concatenate events of different process traces, preserving the chronological order. Based on the concatenation, sub sequences of fixed lengths are for training and prediction. However, the resulting training data is likely to exhibit sequences containing events of different process cases as depicted in Fig. 17. Contrary to the application of LSTMs in Natural Language Processing (NLP), consecutive process cases of an event log exhibit a different relation than sentences in natural language texts. Another approach used in [13] extends sequences exhibiting a shorter length than the maximum length within the given dataset. To obtain sequences of equal length, they extend sequences shorter than the maximum length by adding zero values. Nevertheless, this approach is not able to process sequences that exceed the maximum sequence length in the training dataset. Therefore, we aim at providing a deep learning approach, which is capable of handling sequences of different lengths in prediction scenarios, without having trained several neural networks.

Training Data. To avoid feeding overlapping cases into the neural network, we use a sliding window approach to generate the training data for each case individually. We also use a one-hot encoding as proposed in [13]. Fig. 18 depicts the generation of the training data for a single case. Given a process case containing the chronologically ordered events (e1,e2, … ,e_n) and a window size *k*. Equation 1 describes the generation of the input as well as the corresponding labeled output for *i<n-k*.

$$\bigcup_i \{((e_i, e_{i+1}, \dots , e_{i+k-1}), e_{i+k})\} \tag{4}$$



19	24	17	3	34	4	48	48	12	23	1	77
19	24	17	3	34	4	48	48	12	23	1	77
19	24	17	3	34	4	48	48	12	23	1	77
.		.		.		.		.		.	
19	24	17	3	34	4	48	48	12	23	1	77

Fig. 18. Sliding window approach with k=5

Network Architecture. Convolutional Neural Networks (CNNs) are Feedforward Neural Networks (FNN). In contrast to LSTMs, there is no recurrent flow of data within the FNN. A FNN that incorporates at least one convolutional layer in its architecture is a CNN. CNNs are specialized for processing data that exhibits a grid-like structure. Examples include time-series data like reading a sensor value over time that can be thought of as one-dimensional vector. CNNs have also been used to classify DNA sequences [14]. Convolution incorporates three substantial concepts that can help improving a machine learning system: sparse weights, parameter sharing and equivariance [15]. Furthermore, convolution has properties that allow variable input size. In FNNs, all outputs are connected to all inputs of the consecutive layer. This leads to huge matrix multiplications during the forward run and the training of such a network. A CNN has sparse connectivity between the outputs of a layer and the inputs of its consecutive layer. Since the connections in a feedforward neural network are weights, this is also termed as sparse weights. Sparse weights are obtained by using convolutions with filters that are smaller than the data that is fed into a layer. In general, a sequence can have thousands of elements. In most cases, it is enough to build features from small regions of these elements, since cycles can occur in these sequences. A filter capturing less than five elements is likely to encode important and meaningful features in a sequence. This improves the computational efficiency of such models tremendously. Using the same weights more than once in a function is called weight-sharing. In a FNN, each weight is used exactly once to compute the forward path. Contrary, a CNN learns a set of weights that is used for entire sequences. This does not speed up the training process, but decreases the size of the network architecture, which makes it less demanding in terms of memory. Equivariance of a function f to a transformation T is defined as follows:

$$f(T(x)) = T(f(x)). \tag{5}$$

CNNs are equivariant to translation of the input. Intuitively, when the input changes in some way the output changes in the same. This must not be confused with invariance that is achieved by:

$$f(T(x)) = f(x). \tag{6}$$

To achieve translation invariance, it is common to build units of convolutional layers followed by max-pooling layers. We propose a CNN that uses two consecutive units followed by a convolutional layer with filter size 1 and a softmax layer to predict the next process step. Fig. 17 shows the architecture of our network.

Tab. 10. Network Architecture

Index	Type	Filter Size	Number of Filters	Stride	Padding
1	Conv	9	32	1	keep-size
2	ReLU				
3	MaxPooling	2		1	
4	Conv	3	64	1	keep-size
4	ReLU				
4	MaxPooling	2		1	
5	Conv	1	Number of distinct process steps		keep-size
6	Softmax				

Convolutional layer can process variable length inputs due to parameter sharing. In contrast, fully connected layers have a fixed number of inputs and thus a fixed number of weights. Therefore, these models have to be trained on a specific sequence length. As we are using explicitly no fully connected layers and only convolutional and max pooling layers, our network can process sequences of any length.

7.2 Experimental Settings

We used the 2012 and the 2017 BPI challenge datasets to evaluate our novel approach. In order to compare our approach to [13] we used the proposed data selection consisting of the workflow events with status complete. We implemented our proposed convolutional neural network using Tensorflow. Furthermore, we implemented the proposed network architecture of [13] excluding the parts handling the time prediction. We generated our training data using the case-based sliding window approach. For evaluating the prediction performance of both approaches, we fixed the training and prediction sequences to lengths of 2, 5 and 10. Since the 2017 dataset does not contain sequences of length 11, we are not able to measure the performance for sequences of length 10. We apply a 10-fold cross validation to measure the performance on the proposed datasets. The training and evaluation is conducted using a Titan X GPU.

7.3 Results

Tab. 11 shows the performance of the LSTM architecture in comparison to our convolutional neural network. We were able to achieve at least state of the art performance but using a much lighter and thus computationally much more efficient architecture. As a reference the architecture proposed in [13] incorporates 124,206 trainable parameters. Our architecture incorporates only 6% (8,358) trainable parameters in comparison to the LSTM architecture.

A further advantage of the proposed CNN architecture is its computational efficiency. As a reference we trained both architectures with a batch size of 16 over 16 epochs. We observed that the loss-function started to converge in each case at least after 16 epochs. The training of the LSTM took 301 seconds while the training of the CNN took 47 seconds using the BPI 2012 dataset.

Tab. 11. Results of the experimental evaluation

	BPI 2012 (W complete)			BPI 2017 (W complete)		
sequence	*2*	*5*	*10*	*2*	*5*	*10*
LSTM	0.82	0.85	0.86	0.71	0.63	-
CNN	0.82	0.84	0.84	0.72	0.63	-

8 Answering the Process Owners' Questions

Question 1: The first question is concerned with the throughput times per part of the process. In our analysis, we have identified the two subprocesses that take the most time out of the overall process duration. First, a lot of time is required to complete the application and create corresponding offers, which indicates improvement potentials on the side of the bank. Second, it takes particularly long to collect documents for incomplete applications, which is more likely caused by delays on the customer's side. More details are found in Section 5.3.

Question 2: The second question targets the relationship between the incompleteness of an application and the likelihood with which the corresponding offer is rejected. The process owner stated the initial hypothesis, that customers are more likely to reject an offer if they are confronted with multiple requests for completion. Interestingly, our statistical analysis disproved this hypothesis and revealed that customers are actually significantly more likely to accept the ensuing offer, if their application was originally incomplete. At the same time, the absence of incompleteness leads to negative process outcomes. Section 6.3 describes the details of this analysis.

Question 3: The third question asks how many customers ask for more than one offer and how this influences their likelihood of acceptance. We assessed this question similar to the previous one and found out that there is also a statistically significant association between these variables. Customers who ask for more than one offer are more likely to accept it in the long run. Making multiple offers increases the chances that the applications will end up with the positive process outcome and at the same time decreases the chances that the application will get a negative outcome. Details are also found in Section 6.3.

9 Conclusion

In this report, we relied on combining existing well-established process mining tools and statistical techniques with innovative methods and approaches for process analysis based on artificial intelligence and data mining in order to provide a holistic overview on the provided process data. We wanted to give the process owners an idea of the plethora of opportunities and potentials their data offers to current BPM research. While we used the initial question as an orientation to guide the direction of our analysis, we did not limit ourselves to simply providing answers, but also showed additional results of our available techniques in order to paint a more conclusive picture of the underlying process.

After describing the data provided for this challenge and the tools we used, we have structured this report based on the techniques we have used in each section. We start by analyzing the data using Disco and provide a conclusive analysis based on the capabilities of state-of-the-art process mining tools. In this analysis, we also compare this year's log with the process data of the 2012 BPI challenge. In the following chapter, we compare the two logs using our own tool for Business Process Analysis, the RefMod-Miner, which is able to complement the results of the previous chapter. By clustering the process data vertically based on activity distance, we identify subprocesses in the log, which can be used to further narrow down the time-intensive process parts, which are promising starting points for future optimizations. Analyzing the interdependence of process attributes and outcomes by means of a Chi-square ($\chi 2$), we find out that the loan application process tend to end up with positive outcome if it gets at least one incompleteness status and also that applications are slightly more likely to end up with positive outcome when multiple offers are made. To be able to predict the next events for a running process instance and thus enable managing process cases at runtime, we train a convolutional neural network, achieving a simpler network architecture as well as promising values of accuracy.

Although we did not limit ourselves to the original questions, we are able to provide new insights to the process owners. We identify those subprocesses in the data that are the most time-intensive, disprove the original hypothesis that customers are more likely to accept an offer if they are confronted with more requests for completing their data. Furthermore, we proved statistically that making multiple offers leads to positive process outcomes. In addition, we identify differences and commonalities between today's process and the one of 2012 and predict the next process steps using a new approach to process prediction, which is able to handle variable trace lengths, is a considerable improvement over the state-of-the-art. We are convinced that our approach in this BPI challenge to combine existing process mining techniques with data mining and machine learning offers valuable benefits to the process owners and can assist them in further understanding and improving their process. We would like to express our gratitude and appreciation to the Dutch financial institute for providing their process data for research yet another time and to the BPI committee for once again organizing this unique event. **Acknowledgment.** We gratefully acknowledge the support of NVIDIA for the donation of the GPUs used for this research.

References

1. Van Der Aalst, W.: Process mining. Commun. ACM. 55, 76–83 (2012).
2. Antunes, G., Bakhshandelh, M., Borbinha, J., Cardoso, J., Dadashnia, S., Francescomarino, C. De, Gragoni, M., Fettke, P., Gal, A., Ghidini, C., Hake, P., Khiat, A., Klinkmüller, C., Kuss, E., Leopold, H., Loos, P., Meilicke, C., Niesen, T., Pesquita, C., Peus, T., Schoknecht, A., Sheetrit, E., Sonntag, A., Stickenschmidt, H., Thaler, T., Weber, I., Weidlich, M.: The Process Matching Contest 2015. In: Kolb, J., Leopold, H., and Mendling, J. (eds.) Proceedings of the 6th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA-15), September 3-4, Innsbruck, Austria. Köllen Druck+Verlag GmbH, Bonn (2015).
3. Becker, M., Laue, R.: A comparative survey of business process similarity measures. Comput. Ind. 63, 148–167 (2012).

4. Weidlich, M., Polyvyanyy, A., Mendling, J., Weske, M.: Efficient Computation of Causal Behavioural Profiles Using Structural Decomposition. In: Lilius, J. and Penczek, W. (eds.) Applications and Theory of Petri Nets: 31st International Conference, PETRI NETS 2010, Braga, Portugal, June 21-25, 2010. Proceedings. pp. 63–83. Springer Berlin Heidelberg, Berlin, Heidelberg (2010).

5. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In: Lohmann, N., Song, M., and Wohed, P. (eds.) Business Process Management Workshops: BPM 2013 International Workshops, Beijing, China, August 26, 2013, Revised Papers. pp. 66–78. Springer International Publishing, Cham (2014).

6. de Medeiros, A.K.A., Guzzo, A., Greco, G., van der Aalst, W.M.P., Weijters, A.J.M.M., van Dongen, B.F., Saccà, D.: Process Mining Based on Clustering: A Quest for Precision. In: ter Hofstede, A., Benatallah, B., and Paik, H.-Y. (eds.) Business Process Management Workshops: BPM 2007 International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers. pp. 17–29. Springer Berlin Heidelberg, Berlin, Heidelberg (2008).

7. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis. John Wiley & Sons, Ltd (2011).

8. McHugh, M.L.: The chi-square test of independence. Biochem. medica. 23, 143–149 (2013).

9. Sharpe, D.: Your chi-square test is statistically significant: Now what? Pract. Assessment, Res. Eval. 20, (2015).

10. Goodman, L.A.: Partitioning of chi-square, analysis of marginal contingency tables, and estimation of expected frequencies in multidimensional contingency tables. J. Am. Stat. Assoc. 66, 339–344 (1971).

11. Franke, T.M., Ho, T., Christie, C.A.: The chi-square test: often used and more often misinterpreted. Am. J. Eval. 33, 448–458 (2012).

12. Evermann, J., Rehse, J.-R., Fettke, P.: A Deep Learning Approach for Predicting Process Behaviour at Runtime. In: Dumas, M. and Fantinato, M. (eds.) Proceedings of the 1st International Workshop on Runtime Analysis of Process-Aware Information Systems. International Workshop on Runtime Analysis of Process-Aware Information Systems (PRAISE-2016), located at International Conference on Business Process . Springer (2016).

13. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive Business Process Monitoring with LSTM Neural Networks. In: Dubois, E. and Pohl, K. (eds.) Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings. pp. 477–492. Springer International Publishing, Cham (2017).

14. Giang Nguyen, N., Tran, V.A., Ngo, D.L., Phan, D., Lumbanraja, F.R., Faisal, M.R., Abapihi, B., Kubo, M., Satou, K.: DNA Sequence Classification by Convolutional Neural Network. J. Biomed. Sci. Eng. 9, 280–286 (2016).

15. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016).