Workforce Upskilling: A History-Based Approach for Recommending Unfamiliar Process Activities

Anastasiia Pika(✉) and Moe T. Wynn

Queensland University of Technology, Brisbane, Australia
{a.pika,m.wynn}@qut.edu.au

Abstract. Human resource allocation decisions have a direct impact on the performance of a business process. Many approaches to optimal resource allocation have been proposed in the Business Process Management community. The majority of these approaches aim to optimise process performance; hence, recommend activities to experienced employees. To remain competitive, modern organisations also need to grow the capabilities of their employees and offer them upskilling opportunities. In this article, we propose an approach for recommending unfamiliar activities to employees by comparing their work histories with work histories of other similar employees. The aim of the proposed approach is to put employees on a gradual path of multi-skilling whereby they are provided with an opportunity to perform unfamiliar process activities and thus gain their experience through learning-by-doing. The approach is based on the analysis of process execution data and has been implemented. In the evaluation, we compared recommendations provided by the approach with actual activity executions of different employees recorded in process data. The evaluation demonstrated the effectiveness of the approach for different publicly available event logs and configuration settings.

Keywords: Multi-skilling · Resource allocation · Recommendation · Collaborative filtering · Process execution data

1 Introduction

Organisations are constantly striving to balance the amount of work they need to perform in their business operations with the capacity of their workforce to complete said work. Their goal is to allocate the 'right' kind of work to the 'right' person so that the work is completed in time and is of high quality.

Many approaches to efficiently allocate human resources to work activities have been proposed within the Business Process Management community [3]. Existing resource allocation approaches recommend that a number of criteria are considered; for example, resource experience, preferences, availability, previous performance or compatibility [2]. The majority of resource allocation approaches aim to optimise process performance; hence, they recommend activities to experienced resources. While such approaches may improve the performance of processes, they do not necessarily align with the capability development needs of an organisation and its employees. To maintain its competitive edge, modern organisations need to grow the capabilities of their employees over time [21] and offer them learning and development opportunities on the job [5,10]. In this article, we target the following research question: *how can we recommend process activities that are unfamiliar to employees by analysing their work history recorded in process execution data?*

Our proposed approach is based on the assumption that groups of employees (e.g., in the same role or performing similar activities) follow similar learning paths. These learning paths can be discovered from event logs and can guide the allocation of unfamiliar activities. The approach consists of two main parts: (1) find employees with similar work histories to the employee under consideration; and (2) recommend unfamiliar activities to the employee based on the work histories of similar employees. The approach was implemented and evaluated using multiple event logs. The evaluation demonstrated the performance of the approach for different configurations and data sets.

The proposed approach complements other approaches for resource allocation in the literature. In addition to utilising this approach to grow organisational capacity and enrich the skillsets of employees, the approach can be useful for identifying suitable employees for an activity when experienced employees are not available. Thus, the approach is also applicable in the operational support setting for process-aware information systems.

The rest of the paper is organised as follows. In Sect. 2, we present the preliminaries for the proposed approach. Section 3 details the proposed approach for recommending unfamiliar process activities to employees. Section 4 presents the results from the evaluation conducted with multiple real-life event logs. Section 5 discusses assumptions, limitations and directions for further research. Section 6 summarises the related work and Sect. 7 concludes the paper.

2 Preliminaries

Organisations often use information systems to support execution of their business processes and these systems often record information about process executions in event logs [1]. Such logs may contain information about process instance identifiers, activities performed in the process, timestamps of these activities, resources (i.e., employees) who performed the activities and various data attributes related to the process (e.g., customer type). Our approach is based on the analysis of such event logs. Below, we provide a definition of the event log and specify minimum data required by our approach.

Event Log. Let $\mathcal{E}$ be the set of all events, an *event log* $EL \subseteq \mathcal{E}$ is a set of events. Events can have different attributes and we assume that at least the following attributes are recorded for each event: *activity*, *resource* and *time*. The value of attribute a of event e is denoted as e_a. For example, e_{time} is the timestamp of event $e \in EL$. Let R denote the set of all resources, A denote the set of all activities and T denote the set of all timestamps in event log EL. The sets $\mathcal{E}$, EL, A, R and T are finite and non-empty.

3 Approach

The overall idea of our approach is inspired by collaborative filtering (CF) recommender systems which are widely used in e-commerce to recommend new products or services to users. Collaborative filtering is "the process of filtering or evaluating items through the opinions of other people" [15]. CF recommender systems analyse ratings provided by users to items (e.g., movies or books) and recommend new items to users with similar tastes (e.g., to those who liked similar movies or bought similar books). A recommendation cannot be provided if a user has "tastes so unique that they are not shared by anybody else" [15].

An underlying assumption of our approach is that groups of employees follow similar learning paths in an organisation. Employees learn new activities gradually (e.g., new employees first perform a small set of simple activities and over time they start performing activities that require more experience). Information about activities performed by employees at different times is typically recorded in event logs. The main idea of the approach is to analyse such event log data to identify employees with similar work histories and recommend unfamiliar activities to an employee based on the work histories of similar employees.

Figure 1 depicts an illustrative example which we will use to explain the proposed approach. The figure shows activity executions of seven resources who were active during different periods of time before time t. Let us assume that we would like to recommend a new activity to resource r. In order to recommend a new activity to resource r, we first compare activity execution history of r with histories of other resources (r_1–r_6) to identify similar resources (Algorithm 1). We then analyse activity execution histories of similar resources to recommend

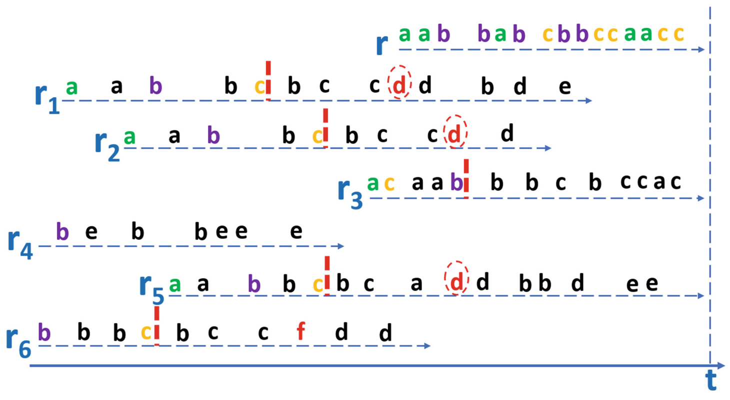
r a a b b a b c b b c c a a c c
r1 a a b b c b c c d d b d e
r2 a a b b c b c c d d
r3 a c a a b b b c b c c a c
r4 b e b b e e e
r5 a a b b c b c a d d b b d e e
r6 b b b c b c c f d d
t

Fig. 1. An illustrative example with activity execution history of seven resources. (Color figure online)

a suitable new activity for resource r (Algorithm 2). Table 1 provides a summary of main steps of Algorithm 1 (Step 1.1–Step 1.3) and Algorithm 2 (Step 2.1–Step 2.2) and shows the output values returned by each step.

Table 1. Output values from Algorithm 1 and Algorithm 2 for the example in Fig. 1.

Algorithm step	Output value
1.1. Extract past activities A^r_{past} for resource r	$A^r_{past} = \{a, b, c\}$
1.2. Identify all similar resources R_{sim}	$R_{sim} = \{r_1, r_2, r_3, r_5, r_6\}$
1.3. Identify similar resources R^{new}_{sim} who performed a new activity	$R^{new}_{sim} = \{r_1, r_2, r_5, r_6\}$
2.1. Identify new activities A_{new} performed by similar resources	$A_{new} = \{d, f\}$
2.2. Recommend new activities A_{rec}	$A_{rec} = \{d\}$

Below, we describe the two algorithms and explain how the values in Table 1 are calculated. The approach takes as input four thresholds:

- $min_{PastActivities}$ – the minimum number of activities that must be performed by a given resource before a recommendation can be provided to the resource.
- $min_{ResourceSimilarity}$ – the minimum resource similarity ([0,1]).
- $min_{SimilarResources}$ – the minimum number of similar resources.
- $min_{ActivitySupport}$ – the minimum fraction of similar resources who must perform a given activity for it to be considered for a recommendation. (For example, if the value of $min_{ActivitySupport}$ is 1 then an activity is only recommended if it was the next new activity performed by all similar resources.)

The value of threshold $min_{PastActivities}$ can be determined based on the knowledge of the process (e.g., a fraction of all activities in the process). The effect of other thresholds on the performance of the approach is evaluated in Sect. 4.

Algorithm 1. Algorithm 1 specifies how we identify similar resources. To identify similar resources, we compare sets of activities performed by different resources. One could also consider other aspects of work history (e.g., activity frequencies or outcomes), we discuss this direction for future work in Sect. 5.

The algorithm takes as input an event log EL, a given resource r (for whom we would like to recommend a new activity), a given time point t (when a recommendation is needed) and thresholds $min_{PastActivities}$, $min_{ResourceSimilarity}$ and $min_{SimilarResources}$. Algorithm 1 consists of three main steps.

In step 1.1, we extract a set of activities A^r_{past} performed by resource r before time t. For the example depicted in Fig. 1, $A^r_{past} = \{a, b, c\}$. If the number of activities in A^r_{past} is lower than the threshold $min_{PastActivities}$, then we cannot provide a recommendation for resource r at time t. The rationale behind this is that if the work history of a resource is very short (e.g., for an employee who just joined an organisation), then there is not enough information to identify similar resources (sufficient information about user history is a requirement of CF recommender systems). Let us assume that $min_{PastActivities} = 2$ for the example in Fig. 1, then the algorithm can proceed (as $|A^r_{past}| > 2$).

In step 1.2, we identify all similar resources R_{sim}. Measuring the similarity of resources based on their complete work histories (e.g., using sequence clustering) would not be appropriate, as the similarity of new employees and experienced employees would be low. Therefore, the similarity $sim^{r\prime}_r$ of resource r and a given resource $r\prime$ is measured as the fraction of activities in A^r_{past} that were also performed by $r\prime$. For example, $sim^{r_1}_r = 1$ as resource r_1 performed all activities in A^r_{past} (i.e., a, b and c); while similarity $sim^{r_6}_r = 2/3$ as resource r_6 only performed activities b and c. A set of similar resources R_{sim} comprises all resources whose similarity with resource r is not lower than threshold $min_{ResourceSimilarity}$. Let us assume that $min_{ResourceSimilarity} = 0.6$ for the example in Fig. 1, then $R_{sim} = \{r_1, r_2, r_3, r_5, r_6\}$ (resource r_4 is not included as $sim^{r_4}_r$ (1/3) is lower than $min_{ResourceSimilarity}$ (0.6)). If the number of similar resources is lower than threshold $min_{SimilarResources}$, then a recommendation cannot be provided. Let us assume that for the example in Fig. 1, $min_{SimilarResources} = 3$, then the algorithm can proceed (as $|R_{sim}| > 3$).

In step 1.3, we identify those similar resources R^{new}_{sim} who performed a new activity after execution of activities in A^r_{past} and before t. First, we identify the earliest time $t^{r\prime}_a$ when all activities from A^r_{past} were performed by a given resource $r\prime$ (the earliest times for similar resources are marked by vertical red dashed lines in Fig. 1). R^{new}_{sim} comprises all similar resources who performed a new activity after time $t^{r\prime}_a$ and before time t. For the example in Fig. 1, $R^{new}_{sim} = \{r_1, r_2, r_5, r_6\}$ (resource r_3 did not perform any new activities). If the number of resources in R^{new}_{sim} is below threshold $min_{SimilarResources}$, then a recommendation is not provided. For the example in Fig. 1, the algorithm can proceed (as $|R^{new}_{sim}| > 3$). Similar resources who did not perform new activities are likely to be at the same stage of their careers as resource r; therefore, we cannot learn from their work histories and they are not further considered.

Algorithm 2. Algorithm 2 recommends the next new activity for resource r at time t. Recommending the next new activity can be challenging for processes in which employees follow similar but not exactly the same learning paths. To better accommodate such scenarios, we also implemented the second version of the approach which recommends a set of new activities (we discuss the second version of the approach at the end of this section). In this paper, we focus on recommending the next new activity or a set of activities; one could also consider recommending activity sequences (i.e., learning paths) or time periods during which a resource should start performing new activities (these directions for future work are discussed in Sect. 5).

The algorithm takes as input event log EL, resource r, time t, threshold $min_{ActivitySupport}$ and outputs of Algorithm 1 (R^{new}_{sim}, $t^{r\prime}_a$ and $A_{t^{r\prime}_a}$ for all $r\prime \in$

Algorithm 1: Find similar resources

```
Input: event log $EL$, resource $r$, time $t$, given thresholds $min_{PastActivities}$,
       $min_{ResourceSimilarity}$ and $min_{SimilarResources}$
Output: $R^{new}_{sim}$, $t^{r\prime}_a$ and $A_{t^{r\prime}_a}$ for all $r\prime \in R^{new}_{sim}$

/* Step 1.1. Extract work history $A^r_{past}$ of resource $r$ at time $t$ */
$A^r_{past} := \{a \in A \mid \exists e \in EL[e_{activity} = a \wedge e_{resource} = r \wedge e_{time} < t]\}$
if $|A^r_{past}| < min_{PastActivities}$ then
    no recommendation
end

/* Step 1.2. Find similar resources $R_{sim}$ */
$R_{sim} := \emptyset$
for each $r\prime \in R$ do
    $A^{r\prime}_{past} := \{a \in A \mid \exists e \in EL[e_{activity} = a \wedge e_{resource} = r\prime \wedge e_{time} < t]\}$
    $sim^{r\prime}_r := |A^r_{past} \cap A^{r\prime}_{past}| / |A^r_{past}|$
    if $r\prime \neq r \wedge sim^{r\prime}_r \geq min_{ResourceSimilarity}$ then
        $R_{sim} := R_{sim} \cup \{r\prime\}$
    end
end
if $|R_{sim}| < min_{SimilarResources}$ then
    no recommendation
end

/* Step 1.3. Find similar resources $R^{new}_{sim}$ who performed a new
   activity after executing all activities in $A^r_{past}$ and before $t$ */
$R^{new}_{sim} := \emptyset$
for each $r\prime \in R_{sim}$ do
    $T_{start} := \emptyset$
    for each $a \in A^r_{past} \cap A^{r\prime}_{past}$ do
        $T^{r\prime}_a := \{t\prime \in T \mid \exists e \in EL[e_{time} = t\prime \wedge e_{resource} = r\prime \wedge e_{activity} = a]\}$
        $T_{start} := T_{start} \cup \{\min_{t\prime \in T^{r\prime}_a} t\prime\}$
    end
    $t^{r\prime}_a := \max_{t\prime \in T_{start}} t\prime$
    $A_{t^{r\prime}_a} := \{a \in A \mid \exists e \in EL[e_{activity} = a \wedge e_{resource} = r\prime \wedge e_{time} < t^{r\prime}_a]\}$
    if $\exists e \in EL[e_{activity} \notin A_{t^{r\prime}_a} \wedge e_{resource} = r\prime \wedge e_{time} > t^{r\prime}_a \wedge e_{time} < t]$ then
        $R^{new}_{sim} := R^{new}_{sim} \cup \{r\prime\}$
    end
end
if $|R^{new}_{sim}| < min_{SimilarResources}$ then
    no recommendation
end
```

R_{sim}^{new}). Our assumption is that the next new activity that will be likely performed by resource r after time t is the next new activity that was frequently performed by similar resources when they were at the same career stage as is resource r at time t. Algorithm 2 consists of two main steps.

In step 2.1, we identify the first new activity performed by each resource $r\prime \in R_{sim}^{new}$ after $t_a^{r\prime}$, a set A_{new} comprises all such activities. In Fig. 1 such activities are highlighted with red color ($A_{new} = \{d, f\}$).

In step 2.2, we recommend new activities A_{rec}. For each activity $a\prime \in A_{new}$, we get activity frequency $freq_{a\prime}$ (the fraction of resources in R_{sim}^{new} who performed the activity); $freq_{max}$ is the maximum activity frequency. In the example in Fig. 1, $freq_d = 0.75$ (as it was the first new activity for three out of four resources), $freq_f = 0.25$ and $freq_{max} = 0.75$. If $freq_{max} < min_{ActivitySupport}$ then a recommendation is not provided (if similar resources perform different new activities then a recommendation is not reliable). Let us assume that for the example in Fig. 1 $min_{ActivitySupport} = 0.6$; hence, a recommendation can be provided (as $freq_{max} > 0.6$). The algorithm returns the set of recommended activities A_{rec} which comprises all activities whose frequency is equal to $freq_{max}$ (in Fig. 1, $A_{rec} = \{d\}$). In practice, A_{rec} usually consists of only one activity, it is possible that A_{rec} includes more than one activity if the process includes activities that are completed at the same time or if $min_{ActivitySupport} \leq 0.5$.

Algorithm 2: RECOMMEND NEW ACTIVITIES

```
Input: event log $EL$, resource $r$, time $t$, threshold $min_{ActivitySupport}$, $R_{sim}^{new}$, $t_a^{r\prime}$
       and $A_{t_a^{r\prime}}$ for all $r\prime \in R_{sim}^{new}$
Output: recommended activities $A_{rec}$
/* Step 2.1. Extract new activities $A_{new}$ from similar resources */
$A_{new} := \emptyset$
for each $r\prime \in R_{sim}^{new}$ do
    $E_{new}^{r\prime} := \{e \in EL \mid e_{time} > t_a^{r\prime} \wedge e_{time} < t \wedge e_{resource} = r\prime \wedge e_{activity} \notin A_{t_a^{r\prime}}\}$
    $T_{new}^{r\prime} := \{t\prime \in T \mid \exists e \in E_{new}^{r\prime}[e_{time} = t\prime]\}$
    $A_{new}^{r\prime} := \{a \in A \mid \exists e \in E_{new}^{r\prime}[e_{activity} = a \wedge e_{time} = \min_{t\prime \in T_{new}^{r\prime}} t\prime]\}$
    $A_{new} := A_{new} \cup A_{new}^{r\prime}$
end
/* Step 2.2. Extract recommended activities $A_{rec}$ */
for each $a \in A_{new}$ do
    $freq_a := |\{r\prime \in R_{sim}^{new} \mid a \in A_{new}^{r\prime}\}| / |R_{sim}^{new}|$
end
$freq_{max} := \max_{a \in A_{new}} freq_a$
if $freq_{max} < min_{ActivitySupport}$ then
    no recommendation
end
$A_{rec} := \{a \in A_{new} \mid freq_a = freq_{max}\}$
```

The algorithms described above recommend the next new activity. In the second version of the approach, we consider for a recommendation all new activities performed by a similar resource within a given time period (e.g., within a week or month) from the first new activity performed by the resource. We then recommend all activities whose frequency is higher than or equal to threshold $min_{ActivitySupport}$ (rather than an activity with the maximum frequency).

4 Evaluation

4.1 Data Sets and Experimental Setup

The performance of the approach was evaluated using three real event logs (Table 2). Two logs, referred to here as BPIC12[1] and BPIC17[2], contain events from a loan application process in a Dutch financial institution recorded during two different time periods. The third log, referred to here as WABO[3], contains information about a receipt phase of an environmental permit application process.

Table 2. Characteristics of event logs used in the evaluation.

Log	Events	Cases	Activities	Resources	Generalists[a]	Log duration
BPIC12	244,190	13,087	24	68	59	166 days
BPIC17	238,481	14,254	25	107	88	6 months[b]
WABO	8,577	1,434	27	48	22	479 days

[a] We refer to resources who perform at least 1/3 of all process activities as generalists.
[b] We used events from the period 1/01/2016–30/06/2016 due to slow performance.

We conducted leave-one-out cross-validation and compared recommendations provided by the approach with actual activity executions recorded in the logs[4]. For a given resource r and for a given point in time t_{split}, a recommendation is learned from events recorded before time t_{split} and compared with the first new activity (or activities for the second version of the approach) recorded for resource r after time t_{split}. This was repeated for all resources in each log. For each resource, time t_{split} was set to the midway between the time of the first activity and the time of the first occurrence of the last new activity recorded for the resource in the log (such split ensures that for each resource at time t_{split} there is at least one activity recorded before and after time t_{split}).

[1] https://data.4tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f.
[2] https://data.4tu.nl/repository/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b.
[3] https://data.4tu.nl/repository/uuid:a07386a5-7be3-4367-9535-70bc9e77dbe6.
[4] The source code is available at https://github.com/a-pika/TaskRecommender.

In the evaluation, we used two performance measures. We measured the fraction of resources for whom a recommendation was provided (referred to here as 'recommendations') and the average accuracy of the recommendations. The first version of our approach recommends the next new activity and the second version of the approach recommends all new activities to be performed within a given time period t_{period} from the first new activity. For the first version, the accuracy of a recommendation provided to a resource at time t_{split} is *1* if the recommended activity is the next new activity recorded for the resource after time t_{split}; otherwise, the accuracy is *0*. For the second version, the accuracy of a recommendation provided to a resource at time t_{split} was measured as the fraction of recommended activities that were performed by the resource after time t_{split} within time t_{period} from the first new activity.

4.2 Results

To evaluate various factors affecting the performance of the approach, we conducted three experiments using the three event logs described in Sect. 4.1. In Experiment 1, we evaluated the effect of different threshold values on the approach performance. In Experiment 2, we compared the performance of the two versions of the approach. In Experiment 3, we compared the performance of the approach for all resources with the performance for generalists only.

Experiment 1: The Impact of Threshold Values on the Performance. In the first experiment, we evaluated the impact of different values of thresholds $min_{ResourceSimilarity}$, $min_{SimilarResources}$ and $min_{ActivitySupport}$ on the performance of the approach. The value of threshold $min_{PastActivities}$ was set to 5 for all event logs and for all experiments[5]. In this experiment, we applied the first version of the approach (i.e., recommending the next new activity). We varied the values of one threshold at a time and fixed the values of two other thresholds. The fixed values used in the experiment were as follows: $min_{ResourceSimilarity} = 0.7$, $min_{SimilarResources} = 3$ and $min_{ActivitySupport} = 0.7$.

Figures 2a–2c show the recommendations and the accuracy for different values of threshold $min_{ResourceSimilarity}$ (ranging from 0.1 to 0.9) for the three logs. We can observe that for log WABO (Fig. 2a), no recommendations were provided for values 0.1 and 0.3 and starting from the minimum resource similarity of 0.5 the accuracy is 1 and the number of recommendations does not change much. For log BPIC12 (Fig. 2b), there are no big differences in the accuracy and the number of recommendations for threshold values 0.1–0.7, while for value 0.9 the accuracy increases and the number of recommendations decreases. For log BPIC17 (Fig. 2c), the accuracy gradually increases, while the number of recommendations does not change much. The charts show that higher values of the minimum resource similarity threshold yield better accuracy; however, the number of recommendations may decrease (due to a smaller number of similar resources). We can also observe that the effect is more pronounced for log

[5] Reliable recommendations cannot be provided to resources with short work histories.

WABO (Fig. 2a) and is less significant for the other two logs. It is possible that there are more similar resources in these two logs; hence, the effect of the minimum resource similarity threshold is less pronounced for them.

Figures 2d–2f demonstrate the effect of threshold $min_{SimilarResources}$ and Figs. 2g–2i show the effect of threshold $min_{ActivitySupport}$ on the approach performance for the three logs. For both thresholds, we can observe similar trends for the three logs: higher values of a threshold ($min_{SimilarResources}$ or $min_{ActivitySupport}$) yield better accuracy and reduce the number of recommendations. Figure 2 shows that threshold $min_{ActivitySupport}$ has the biggest impact on the performance for all logs. It is expected that the accuracy is poor for values of $min_{ActivitySupport}$ that are below 0.5 as in such scenarios more than one activity can be included in the recommendation (when similar resources perform different unfamiliar activities), and hence, the recommendation is not reliable.

Experiment 1 showed the trade-off between the accuracy and the number of recommendations. For example, in Fig. 2h, recommendations are provided to 41% of resources with the average accuracy of 0.61; however, they could only be provided to 3% of resources with the accuracy of 1. The experiment also showed that the impact of the thresholds on the performance is similar for all logs (higher values yield better performance); however, specific values that could achieve a

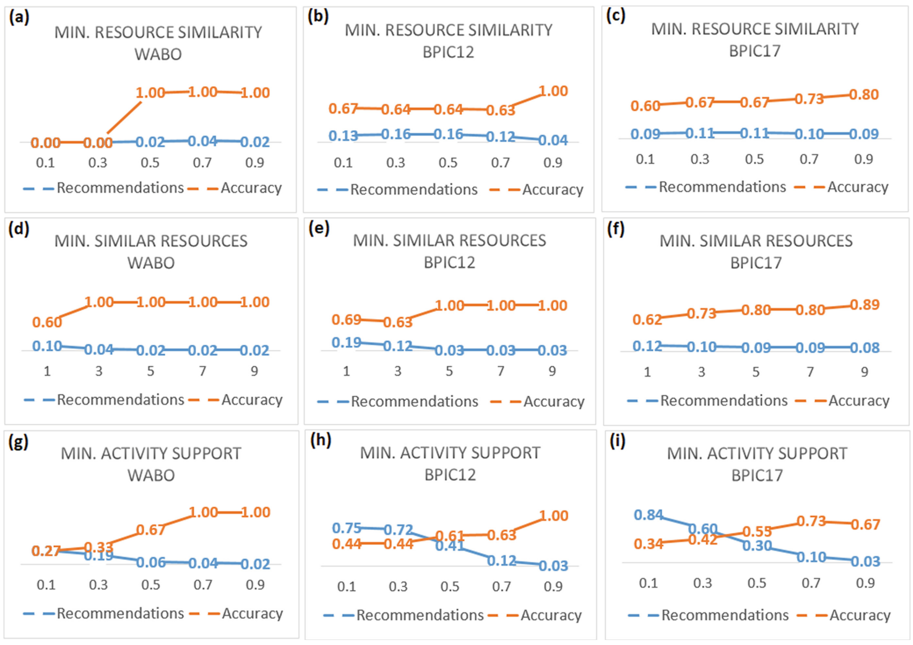

Fig. 2. Experiment 1: the impact of threshold values on the approach performance (Recommendations – the fraction of resources for whom a recommendation was provided; accuracy – the average accuracy of the recommendations).

given level of performance are different for different logs. In future work, we plan to use machine learning to configure the thresholds.

Experiment 2: The Performance of Different Versions of the Approach. In Experiment 1, we evaluated the first version of our approach which recommends the next new activity. In Experiment 2, we evaluated the second version of the approach which recommends a set of new activities to be performed within a given period of time from the next new activity. We evaluated two time periods: 7 days and 30 days. We used the following values of the thresholds: $min_{ResourceSimilarity} = \{0.6, 0.7, 0.8\}$, $min_{SimilarResources} = \{2, 3, 4\}$ and $min_{ActivitySupport} = \{0.6, 0.7, 0.8\}$ (the values were selected based on the results of Experiment 1 as they yielded optimal performance for the three logs). The experiment was performed for all combinations of these values (i.e., 27 times) and we report the average values in Fig. 3. Figure 3a shows the average fraction of resources for whom a recommendation was provided (by two versions of the approach) and Fig. 3b shows the average accuracy of the recommendations. We can see that for all logs the number of recommendations is higher for the second version of the approach and is higher for the longer time period (i.e., 30 days). The accuracy does not change much for log WABO and for the other two logs it is higher for the second version of the approach. The experiment shows that predicting the next new activity is more challenging than predicting a set of new activities and the longer the time period that is considered, the better the performance of the approach. This outcome is expected as the number of employees who follow similar learning paths will be usually higher than the number of employees who follow the exact same learning path in an organisation.

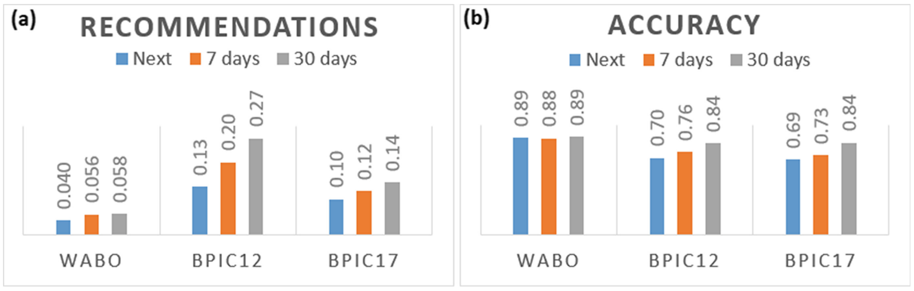

Fig. 3. Experiment 2: the performance of different versions of the approach (Next – recommending the next new activity; 7 days and 30 days – recommending new activities to be performed within 7 days and 30 days from the next new activity).

Experiment 3: The Performance of the Approach for Generalists. Experiments 1 and 2 were conducted using information about all resources recorded in the event logs. This approach may not be suitable for resources who only perform very few activity types in the process (i.e., specialists). In Experiment 3, we compared the performance of the approach for all resources with the performance for generalists. We set the threshold that generalists perform at least 1/3 of all activity types in the process. Table 2 shows the number of all resources and the number of generalists in each log. We used the same threshold values as in Experiment 2 and we show the average performance values in Fig. 4. We evaluated the performance of both versions of the approach (denoted as ‘Next’ and ‘30 days’ in Fig. 4). As expected, the number of recommendations is higher for generalists than for all resources (Fig. 4a) for both versions of the approach for all event logs, while the average accuracy does not change much (Fig. 4b).

The experiments demonstrated the performance of the approach for different configurations and real event logs. They showed that the approach can provide recommendations to multiple resources with the accuracy ranging from 0.6 to 0.9 (depending on the configuration and the log, see Figs. 2–4), while few recommendations can be provided with the accuracy close to 1.

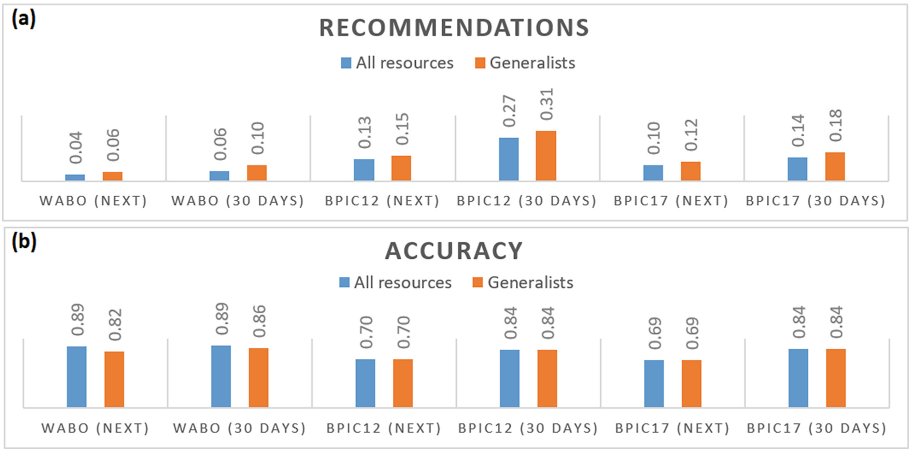

Fig. 4. Experiment 3: the performance of the approach for all resources and generalists.

5 Assumptions, Limitations and Future Work

An assumption of our approach is that groups of employees follow similar learning paths; the proposed approach is not suitable for organisations in which employees have unique career trajectories. We assume that there are no dependencies between recommended activities and activities previously performed by a resource; an investigation of activity pre-requisites is a direction for future work. We also assume that the process is in a steady state (i.e., it is not changing over time); one could investigate the possibility to only use recent work history.

A limitation of our approach is the use of a simple resource similarity measure which compares sets of activities performed by different employees. A direction for future work is to devise resource similarity measures that consider additional factors (e.g., the order of activities, activity frequencies, durations or activity outcomes [13]). A richer notion of resource similarity could help to improve the performance of the approach (it could be possible to provide more accurate recommendations with fewer similar resources). Another limitation of the approach is the need to specify the values of the four thresholds which are input to the approach; in future work we plan to configure the thresholds (e.g., by using hyperparameter optimisation techniques). The performance of the approach was evaluated by comparing activity recommendations with actual activity executions recorded in event logs; a direction for future work is an evaluation of the effect of the approach on organisational capacity and employee development.

The approach presented in this article is algorithmic; one could also consider the possibility to use machine learning techniques. For example, an application of clustering algorithms to identify similar resources could be investigated; a challenge is that learning paths of new and experienced employees may not be considered similar by such algorithms. One could also apply machine learning algorithms (e.g., deep neural networks) to learn activity recommendations; however, such algorithms often require large training sets, and hence, may only be suitable for large organisations.

In this work, we focused on recommending unfamiliar activities to potential generalists; the approach could be extended for specialists to recommend unfamiliar case or activity types (e.g., recommending more complex cases or activities related to new product groups). Another direction for future work is recommending learning paths (i.e., activity sequences rather than the next new activity). One could also learn from history suitable time periods for new activities (e.g., a new activity should be performed no later than six months from now). Finally, the possibility to combine the proposed approach with approaches that consider other resource allocation criteria (e.g., resource compatibility or previous performance) [2] could be investigated.

6 Related Work

Business Process Management is concerned with the development of new methods and techniques for design, implementation, execution and analysis of business processes. Employees and their skill sets play a crucial role in running efficient and cost effective business operations and organisations are always exploring ways to grow the capabilities of their employees over time (e.g., a learning organisation [5]). Process-aware information systems allocate resources to process activities at run time according to some pre-defined criteria (e.g., based on roles); however, “current systems provide limited support for resource allocation” and “actual resource allocation is delegated to people to some extent” [7]. Systems that support flexible business processes allow employees to select methods suitable for a particular process instance (rather than allocating specific tasks to resources) [4,6]; however, resources who can handle the process instance are still selected based on pre-defined criteria (e.g., roles or qualifications [4]).

Russell et al. [14] proposed a set of resource patterns that describe how a resource interacts with a process-aware information system. A role-based allocation pattern, for example, offers the work to a resource with a specific role while a round-robin allocation pattern offers a fair and equitable allocation by sharing work equally among resources. Other patterns utilise the knowledge of previous performance of resources (e.g., in the case of the retain familiar pattern and the history-based distribution pattern). Although these resource patterns do not explicitly consider workforce upskilling concerns, the proposed approach can be seen as an advanced form of the history-based distribution pattern [14].

Organisational mining is an area of process-oriented data mining which is concerned with extracting insights about various aspects of resource behaviour from event logs [13,19] and includes history-based resource allocation approaches [2,3]. A recent systematic mapping study [3] identified 95 approaches which tackle the problem of human resource allocation in business processes. A taxonomy of human resource allocation criteria was proposed by Arias et al. [2] based on an extensive literature review of existing approaches. The proposed taxonomy captures the following resource allocation criteria: the number of required resources, resource experience and expertise, resource preferences, previous performance, role, social context (e.g., collaboration or compatibility), trustworthiness ("notion of trust degree that a resource may have to execute activities") and resource workload [2]. These works [2,3] provide an extensive overview of resource allocation approaches; however, *the capability development needs of employees (i.e., workforce upskilling) are not explicitly considered by the taxonomy.*

Existing resource allocation approaches can discover resource assignment rules from event logs (e.g., Schönig et al. [16,17]); optimise resource allocation in order to improve process performance (e.g., an approach proposed by Park and Song [12] optimises resource allocation for "a maximum flow with the smallest possible cost"; an approach proposed by Huang et al. [9] optimises resource allocation "by trying to minimize long-term cost" using reinforcement learning; and an approach proposed by Havur et al. [8] derives "an optimal schedule for work items that have dependencies"); or they provide support for resource allocation decisions (e.g., a recommender system proposed by Sindhgatta et al. [18] "uses information on the performance of similar resources in similar contexts to predict a resource's suitability for a task"). *All these approaches are process-centric (i.e., they aim to improve the performance of a process) and they disregard skill development needs of employees.*

The need to take human-centric approach in resource allocation was raised by Kabicher-Fuchs and Rinderle-Ma [11] who proposed a collection of work experience measures. Kabicher-Fuchs et al. [10] proposed a resource allocation algorithm which considers experience development goals explicitly specified by employees. A genetic algorithm based approach by Starkey et al. [20] optimises team moves and upskilling specifically for engineers; "the next logical skill set for any given engineer" is an input to the algorithm. Similar to these works, we argue that resource development needs should be considered by resource allocation methods; however, unlike these works, *our approach recommends new activities to employees by analysing their work history recorded in event logs.*

7 Conclusion

Human resource allocation decisions have a direct impact on the performance of business processes. The problem of assigning suitable employees to process activities has got a lot of attention in the Business Process Management community; however, the majority of resource allocation approaches do not consider the capacity development needs of organisations and individual employees. In this article, we proposed an approach for recommending unfamiliar process activities to employees by analysing work histories recorded in process execution data. The approach was implemented and evaluated using three real event logs. The evaluation demonstrated the effectiveness of the approach for different event logs and configuration settings. The proposed approach is the first attempt to provide history-based recommendations of unfamiliar process activities to employees using process execution data and it opens an array of opportunities for further research, which we discussed in this article.

References

1. van der Aalst, W.: Process Mining: Data Science in Action. Springer, Berlin (2016). https://doi.org/10.1007/978-3-662-49851-4_1
2. Arias, M., Munoz-Gama, J., Sepúlveda, M.: Towards a taxonomy of human resource allocation criteria. In: Teniente, E., Weidlich, M. (eds.) BPM 2017. LNBIP, vol. 308, pp. 475–483. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74030-0_37
3. Arias, M., Saavedra, R., Marques, M.R., Munoz-Gama, J., Sepúlveda, M.: Human resource allocation in business process management and process mining: a systematic mapping study. Manag. Decis. **56**(2), 376–405 (2018)
4. Dellen, B., Maurer, F., Pews, G.: Knowledge-based techniques to increase the flexibility of workflow management. Data Knowl. Eng. **23**(3), 269–295 (1997)
5. Dymock, D., McCarthy, C.: Towards a learning organization? employee perceptions. Learn. Organ. **13**(5), 525–537 (2006)
6. Faustmann, G.: Enforcement vs freedom of action an integrated approach to flexible workflow enactment. ACM SIGGROUP Bull. **20**(3), 5–6 (1999)
7. Havur, G., Cabanillas, C.: History-aware dynamic process fragmentation for risk-aware resource allocation. In: Panetto, H., Debruyne, C., Hepp, M., Lewis, D., Ardagna, C.A., Meersman, R. (eds.) OTM 2019. LNCS, vol. 11877, pp. 533–551. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33246-4_33
8. Havur, G., Cabanillas, C., Mendling, J., Polleres, A.: Resource allocation with dependencies in business process management systems. In: La Rosa, M., Loos, P., Pastor, O. (eds.) BPM 2016. LNBIP, vol. 260, pp. 3–19. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45468-9_1
9. Huang, Z., van der Aalst, W.M., Lu, X., Duan, H.: Reinforcement learning based resource allocation in business process management. Data Knowl. Eng. **70**(1), 127–145 (2011)
10. Kabicher-Fuchs, S., Mangler, J., Rinderle-Ma, S.: Experience breeding in process-aware information systems. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 594–609. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38709-8_38

11. Kabicher-Fuchs, S., Rinderle-Ma, S.: Work experience in PAIS – concepts, measurements and potentials. In: Ralyté, J., Franch, X., Brinkkemper, S., Wrycza, S. (eds.) CAiSE 2012. LNCS, vol. 7328, pp. 678–694. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31095-9_44
12. Park, G., Song, M.: Prediction-based resource allocation using LSTM and minimum cost and maximum flow algorithm. In: ICPM 2019, pp. 121–128. IEEE (2019)
13. Pika, A., Leyer, M., Wynn, M.T., Fidge, C.J., ter Hofstede, A., van der Aalst, W.: Mining resource profiles from event logs. ACM TMIS **8**(1), 1 (2017)
14. Russell, N., Van Der Aalst, W.M., Ter Hofstede, A.H.: Workflow Patterns: The Definitive Guide. MIT Press, Cambridge (2016)
15. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web. LNCS, vol. 4321, pp. 291–324. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72079-9_9
16. Schönig, S., Cabanillas, C., Di Ciccio, C., Jablonski, S., Mendling, J.: Mining team compositions for collaborative work in business processes. Softw. Syst. Model. **17**(2), 675–693 (2016). https://doi.org/10.1007/s10270-016-0567-4
17. Schönig, S., Cabanillas, C., Jablonski, S., Mendling, J.: A framework for efficiently mining the organisational perspective of business processes. Decis. Support Syst. **89**, 87–97 (2016)
18. Sindhgatta, R., Ghose, A., Dam, H.K.: Context-aware recommendation of task allocations in service systems. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 402–416. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_25
19. Song, M., van der Aalst, W.: Towards comprehensive support for organizational mining. Decis. Support Syst. **46**(1), 300–317 (2008)
20. Starkey, A.J., Hagras, H., Shakya, S., Owusu, G.: A genetic algorithm based approach for the simultaneous optimisation of workforce skill sets and team allocation. In: Bramer, M., Petridis, M. (eds.) Research and Development in Intelligent Systems XXXIII, pp. 253–266. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47175-4_19
21. Ulrich, D., Lake, D.: Organizational capability: creating competitive advantage. Acad. Manag. Perspect. **5**(1), 77–92 (1991)